

Handed out: 11/07/2015

Due by 11:59 PM on Friday, 11/13/2015

Place all of your narratives and illustrations in a single Word or PDF document named E90_LastNameFirstNameHW10.docx [.pdf]. Use this assignment as the initial template. Please implement code solutions as a separate class in one (Java, C#, Ruby, Python,...) project. Add your steps and your code below problem statements used for each problem. Upload your homework file and your working code (e.g., Filename.java) into your Assignment 10 folder. Do not include executables. Please also do not zip your files.

This assignment requires that you have a working copy of AMI you created in your solution for Problem 3 of Assignment 09. That AMI needs to have `index.php` or an equivalent file, if you are working in a language different than Java, installed and working. That AMI needs to have `serversideconsumer.jar` executable jar (or its equivalent in another language) installed and functioning properly. Finally, on the client side you need to have a working copy of `clientsideproducer.java` or its equivalent in another language.

You do not have to document creation of the above AMI.

Problem 1:

Please run command line instructions referred to in this assignment using AWS command line interface.

Create an Auto Scaling launch configuration. Use a command that approximately reads like this:

```
C:\> aws autoscaling create-launch-configuration --launch-configuration-name sat_lc --image-id ami-adadd2c7 --instance-type t1.micro --key-name ec2_hu --security-groups default
```

Please consult AWS documentation for the proper syntax of every AWS API command you plan to use. Our launch configuration is called `sat_lc`. `ami-adadd2c7` is the ID of the AMI we created. You should use your own. `ec2_hu` is the name of our key pair file. Use yours. Our `default` group has ports 22 and 80 and CID of `0.0.0.0/0`. Check your `default` group and make sure it has ports 22 and 80 open. You are defining the security group that will control access to instances in your future auto scaling group.

1) Create the auto scaling launch configuration.

```
hqiu@bos-mpdei>> aws autoscaling create-launch-configuration --launch-configuration-name as_lc --image-id ami-24443d4e --instance-type t2.micro --key-name ec2hqiu --security-groups launch-hqiu
```

```
hqiu@bos-mpdei>> aws autoscaling describe-launch-configurations --launch-configuration-names "as_lc"
```

```
{
    "LaunchConfigurations": [
        {
            "UserData": "",
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn:aws:autoscaling:us-east-1:217134905396:launchConfiguration:ec1368c9-c13e-48db-a899-2910b230054f:launchConfigurationName/as_lc",
            "InstanceMonitoring": {
                "Enabled": true
            },
            "ClassicLinkVPCSecurityGroups": [],
            "CreatedTime": "2015-11-11T22:21:15.890Z",
            "BlockDeviceMappings": [],
            "KeyName": "ec2hqiu",
            "SecurityGroups": [
                "launch-hqiu"
            ],
            "LaunchConfigurationName": "as_lc",
            "KernelId": "",
            "RamdiskId": "",
            "ImageId": "ami-24443d4e",
            "InstanceType": "t2.micro"
        }
    ]
}
```

The security group “launch-hqiu” has port 22 and 80 open.

The screenshot shows the AWS Lambda console interface. At the top, there are three buttons: 'Create launch configuration' (highlighted in blue), 'Create Auto Scaling group', and 'Actions'. Below this is a search bar labeled 'Filter: Filter launch configurations...'. A table lists one launch configuration:

Name	AMI ID	Instance Type	Creation Time
as_lc	ami-24443d4e	t2.micro	November 11, 2015 5:21:15 PM...

Below the table, it says 'Launch Configuration: as_lc'. Under the 'Details' tab, the configuration parameters are listed:

AMI ID	ami-24443d4e	Instance Type	t2.micro
IAM Instance Profile		Kernel ID	
Key Name	ec2hqiu	Monitoring	true
EBS Optimized	false	Security Groups	launch-hqiu

Next create an auto scaling group based on the above launch configuration. Use a AWS CLI command similar to the following:

```
C:\> aws autoscaling create-auto-scaling-group --auto-scaling-group-name sat_asg --launch-configuration-name sat_lc --availability-zones us-east-1d, us-east-1b --min-size 2 --max-size 10 --desired-capacity 4 --load-balancer-names SaturdayLB --health-check-type ELB --health-check-grace-period 500
```

Our auto scaling group is called `sat_asg`. It relies on previously created launch configuration `sat_lc`. You can have all instances in your auto scaling group in one availability zone. You can choose several different zones. Our load balancer was called

SaturdayLB. Yours could be called SundayLB, or whatever you choose. Health check type ELB specifies that the auto scaling group will rely on the load balancer's assessment of the health of instances in the group. If grace period option is still there, it tells the autoscaling group that it does not have to recheck its compliance with conditions, more frequently than once in 50 seconds. You can change that value up or down.

2) Create the auto scaling group.

```
hqiu@bos-mpdei>> aws autoscaling create-auto-scaling-group --auto-scaling-group-name my_asg --launch-configuration-name as_lc --availability-zones us-east-1b us-east-1c --min-size 2 --max-size 10 --desired-capacity 4 --load-balancer-names myLB --health-check-type ELB --health-check-grace-period 100
```

It relies on the launch configuration “as_lc” and load balancer “myLB”.

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	4	4	2	10	us-east-1c, us-east-1b	300	100

Once the auto scaling group is created, it will launch the desired number of instances. The load balancer will help check the health status of the instances.

Create Auto Scaling group Actions ▾

1 to 1 of 1 Auto Scaling Group								
	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown
	my_asg	as_lc	4	4	2	10	us-east-1c, us-east-1b	300

Auto Scaling Group: my_asg

Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status ▾ Q Filter scaling history... ▾ 1 to 4 of 4 History Items

Status	Description	Start Time	End Time
Successful	Launching a new EC2 instance: i-9e05a12e	2015 November 11 17:26:25 UTC-5	2015 November 11 17:27:01 UTC-5
Successful	Launching a new EC2 instance: i-9f05a12f	2015 November 11 17:26:25 UTC-5	2015 November 11 17:26:59 UTC-5
Successful	Launching a new EC2 instance: i-30367387	2015 November 11 17:26:24 UTC-5	2015 November 11 17:27:27 UTC-5
Successful	Launching a new EC2 instance: i-31367386	2015 November 11 17:26:24 UTC-5	2015 November 11 17:27:29 UTC-5

Create Auto Scaling group Actions ▾

Filter: Q Filter Auto Scaling groups... ▾ 1 to 1 of 1 Auto Scal

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health C
my_asg	as_lc	4	4	2	10	us-east-1c, us-east-1b	300	100

Auto Scaling Group: my_asg

Details Activity History Scaling Policies Instances Notifications Tags

Actions ▾

Filter: Any Health Status ▾ Any Lifecycle State ▾ Q Filter instances... ▾ 1 to 4 of 4 Ins

Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Sta
i-30367387	InService	as_lc	us-east-1b	Healthy
i-31367386	InService	as_lc	us-east-1b	Healthy
i-9e05a12e	InService	as_lc	us-east-1c	Healthy
i-9f05a12f	InService	as_lc	us-east-1c	Healthy

Filter: Q Search Load Balancers ▾

Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count
myLB	myLB-221011144.us-east-1....	80 (HTTP) forwarding to 80 (...)	us-east-1a, us-east-1b,...	4 Instances

Load balancer: myLB

Description Instances Health Check Monitoring Security Listeners Tags

DNS Name: myLB-221011144.us-east-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see [Using Domain Names With Elastic Load Balancing](#).

Scheme: internet-facing

Status: 4 of 4 instances in service

Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)
Stickiness: Disabled (Edit)

We need two policies, one will specify how many instances we will add to the group when we want to increase the number of instances and the other will specify how many we will terminate when we want to decrease the number of instances. In our case, the increment will be 1 in both directions. In practice you can have different values. Auto scaling policies are created with the AWS API command `put-scaling-policy`, like this:

```
C:\> aws autoscaling put-scaling-policy --auto-scaling-group-name sat_asg --policy-name scale-up --scaling-adjustment 3 --adjustment-type ChangeInCapacity --cooldown 50
```

We created two policies with respective names `scale-up` and `scale-dn`. You can call your policies differently. The `cooldown` parameter is similar to the grace period above and does not have to be the same on the way up and on the way down.

If you forget which policies you have and would like to remind yourself how they are set up you could issue: `aws autoscaling describe-policies` command.

3) Create two auto-scaling policies.

```
hqiu@bos-mpdei>> aws autoscaling put-scaling-policy --auto-scaling-group-name my_asg --policy-name scale-up --scaling-adjustment 3 --adjustment-type ChangeInCapacity --cooldown 50
{
    "PolicyARN": "arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ced14c30-e0b0-46f8-8c45-5f8214b6a210:autoScalingGroupName/my_asg:policyName/scale-up"
}
hqiu@bos-mpdei>> aws autoscaling put-scaling-policy --auto-scaling-group-name my_asg --policy-name scale-dn --scaling-adjustment -2 --adjustment-type ChangeInCapacity --cooldown 50
{
    "PolicyARN": "arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:edff66d5-94ef-4519-b079-4259e25ecad5:autoScalingGroupName/my_asg:policyName/scale-dn"
}
```

```
hqiu@bos-mpdei>> aws autoscaling describe-policies
{
    "ScalingPolicies": [
        {
            "PolicyName": "scale-dn",
            "AutoScalingGroupName": "my_asg",
            "PolicyARN": "arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:edff66d5-94ef-4519-b079-4259e25ecad5:autoScalingGroupName/my_asg:policyName/scale-dn",
            "Cooldown": 50,
            "PolicyType": "SimpleScaling",
            "StepAdjustments": [],
            "AdjustmentType": "ChangeInCapacity",
            "Alarms": [],
            "ScalingAdjustment": -2
        },
    ]
}
```

```
{
    "PolicyName": "scale-up",
    "AutoScalingGroupName": "my_asg",
    "PolicyARN": "arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ced14c30-e0b0-46f8-8c45-5f8214b6a210:autoScalingGroupName/my_asg:policyName/scale-up",
    "Cooldown": 50,
    "PolicyType": "SimpleScaling",
    "StepAdjustments": [],
    "AdjustmentType": "ChangeInCapacity",
    "Alarms": [],
    "ScalingAdjustment": 3
}
]
```

Auto Scaling Group: my_asg

Scaling Policies

Add policy

scale-dn

Execute policy when: No alarm selected

Take the action: Remove 2 instances

And then wait: 50 seconds before allowing another scaling activity

scale-up

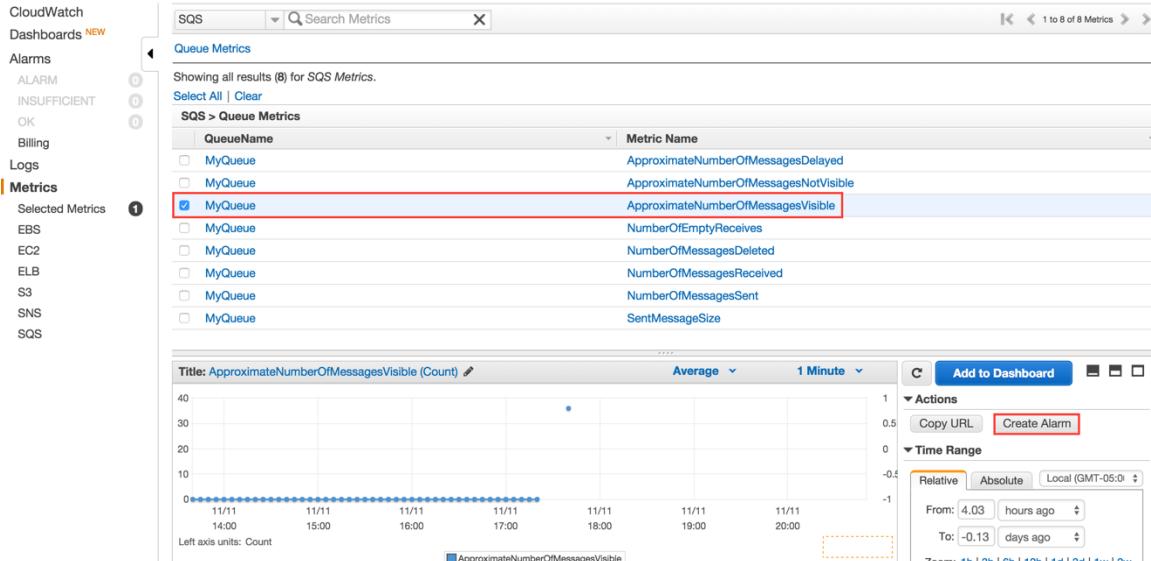
Execute policy when: No alarm selected

Take the action: Add 3 instances

And then wait: 50 seconds before allowing another scaling activity

Setup 2 CloudWatch alarms using SQS metric `ApproximateNumberOfMessagesVisible`. One of them should trigger a “down” policy, the other the “up” policy.

- 4) Create two alarms based on the SQS metric. The “UpwardMovingSQSAlarm” fires when the “`ApproximateNumberOfMessagesVisible`” ≥ 200 . The “DownwardMovingSQSAlarm” fires when “`ApproximateNumberOfMessagesVisible`” ≤ 50 .



Create Alarm

1. Select Metric **2. Define Alarm**

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: UpwardMovingSQSAlarm

Description: UpwardMovingSQSAlarm

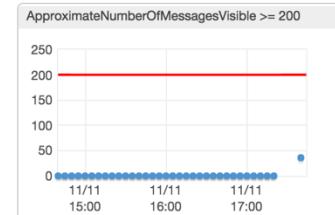
Whenever: ApproximateNumberOfMessagesVisible

is: \geq 200

for: 2 consecutive period(s)

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 2 minutes



Actions

Define what actions are taken when your alarm changes state.

Notification	Delete
Whenever this alarm: State is ALARM	
Send notification to: ASTopic	New list Enter list
Email list: hanjiaoqiu@g.harvard.edu	

+ Notification **+ AutoScaling Action** + EC2 Action

Namespace: AWS/SQS

QueueName: MyQueue

Metric Name: ApproximateNumberof

Period: 1 Minute

Statistic: Average

Cancel Previous Next **Create Alarm**

AutoScaling Action	Delete
Whenever this alarm: State is ALARM	
From the group: my_asg	
Take this action: scale-up - Add 3 instances	

+ Notification + AutoScaling Action + EC2 Action

Cancel Previous Next **Create Alarm**

Create Alarm

[1. Select Metric](#) [2. Define Alarm](#)

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: DownwardMovingSQSAlarm

Description: DownwardMovingSQSAlarm

Whenever: ApproximateNumberOfMessagesVisible
is: <= 50
for: 2 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification

Whenever this alarm: State is ALARM
Send notification to: ASTopic
Email list: hanjiaoqiu@g.harvard.edu

AutoScaling Action

Whenever this alarm: State is ALARM

AutoScaling Action

Whenever this alarm: State is ALARM
From the group: my_asg
Take this action: scale-dn - Remove 2 instances

Alarm Preview

This alarm will trigger when the blue line goes down to or below the red line for a duration of 2 minutes

Namespace: AWS/SQS
QueueName: MyQueue
Metric Name: ApproximateNumberOfMessagesVisible

Period: 1 Minute
Statistic: Average

[Cancel](#) [Previous](#) [Next](#) [Create Alarm](#)

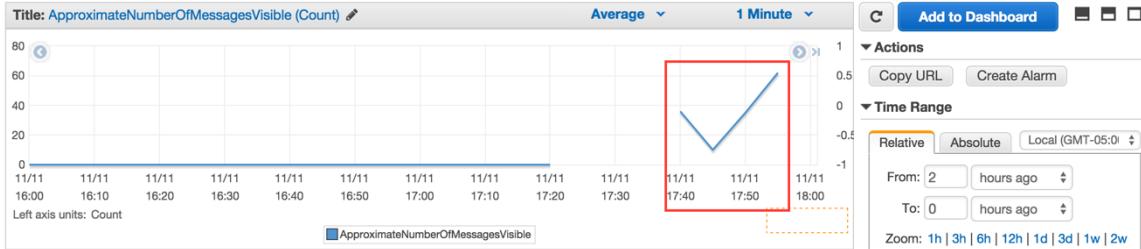
[+ Notification](#) [+ AutoScaling Action](#) [+ EC2 Action](#)

[Cancel](#) [Previous](#) [Next](#) [Create Alarm](#)

The alarm will send notification to topic of the subscriber and take the auto scale actions.

Test several message production speeds and several values of up threshold and down threshold. Convince yourself that auto scaling will kick in and start increasing the number of instances if the number of messages in the queue starts piling up beyond the upward threshold. Convince yourself that auto scaling will start terminating instances once the number of messages in your queue drops below the downward threshold. Verify that instances under the load balancer is adjusted dynamically to the number of instances in the auto scaling group. Be aware that CloudWatch reports slowly, with a 5 minute latency, on the status of your queue. This latency makes these processes somewhat slow.

- 5) Verify the auto scaling will scale the instances dynamically.



CloudWatch Alarms

ALARM

INSUFFICIENT

OK

Billing

Logs

Metrics

Selected Metrics

EBS
EC2
ELB
S3
SNS
SQS

Create Alarm **Modify** **Copy** **Delete**

Filter: State is ALARM **Search Alarms**

State	Name	Threshold	Config Status
ALARM	DownwardMovingSQSAlarm	ApproximateNumberOfMessagesVisible <= 50 for 2 minutes	

1 Alarm selected

Alarm: DownwardMovingSQSAlarm

Details **History**

State Details: State changed to ALARM at 2015/11/11. Reason: Threshold Crossed: 1 datapoint (35.0) was less than or equal to the threshold (50.0).

Description: DownwardMovingSQSAlarm

Threshold: ApproximateNumberOfMessagesVisible <= 50 for 2 minutes

Actions: In ALARM: For group my_asg use policy scale-dn (Remove 2 instances)
Send message to topic "ASTopic" (hanjiaoqu@g.harvard.edu)

Namespace: AWS/SQS
Metric Name: ApproximateNumberOfMessagesVisible
Dimensions: QueueName = MyQueue
Statistic: Average
Period: 1 minute

DownwardMovingSQSAlarm

ApproximateNumberOfMessagesVisible <= 50

At the beginning, the “ApproximateNumberOfMessagesVisible” in the queue was less than 50. The “DownwardMovingSQSAlarm” was fired. The alarm will notify the subscriber and trigger the scale down policy and remove 2 instances.

ALARM: "DownwardMovingSQSAlarm" in US - N. Virginia

Inbox

ASTopic <no-reply@sns.amazonaws.com> to me 5:54 PM (0 minutes ago)

You are receiving this email because your Amazon CloudWatch Alarm "DownwardMovingSQSAlarm" in the US - N. Virginia region has entered the ALARM state, because "Threshold Crossed: 1 datapoint (35.0) was less than or equal to the threshold (50.0)." at "Wednesday 11 November, 2015 22:54:22 UTC".

View this alarm in the AWS Management Console: <https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#s=Alarms&alarm=DownwardMovingSQSAlarm>

Alarm Details:

- Name: DownwardMovingSQSAlarm
- Description: DownwardMovingSQSAlarm
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint (35.0) was less than or equal to the threshold (50.0).
- Timestamp: Wednesday 11 November, 2015 22:54:22 UTC
- AWS Account: 217134905396

Threshold:
- The alarm is in the ALARM state when the metric is LessThanOrEqualToThreshold 50.0 for 60 seconds.

Monitored Metric:

- MetricNamespace: AWS/SQS
- MetricName: ApproximateNumberOfMessagesVisible
- Dimensions: [QueueName = MyQueue]
- Period: 60 seconds
- Statistic: Average
- Unit: not specified

State Change Actions:

- OK:
- ALARM: [arn:aws:sns:us-east-1:217134905396:ASTopic] [arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:edff66d5-94ef-4519-b079-4259e25ecad5:autoScalingGroupName/my_asg:policyName/scale-dn]
- INSUFFICIENT_DATA:

Auto Scaling Group: my_asg

Instances

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	4	2	2	10	us-east-1c, us-east-1b	300	100

Auto Scaling Group: my_asg

Instances

Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status
i-30367387	Terminating	as_lc	us-east-1b	Healthy
i-31367386	InService	as_lc	us-east-1b	Healthy
i-9e05a12e	InService	as_lc	us-east-1c	Healthy
i-9f05a12f	Terminating	as_lc	us-east-1c	Healthy

Auto Scaling Group: my_asg

Activity History

Status	Description	Start Time	End Time
Successful	Terminating EC2 instance: i-9f05a12f	2015 November 11 17:54:51 UTC-5	2015 November 11 17:56:07 UTC-5
Successful	Terminating EC2 instance: i-30367387	2015 November 11 17:54:50 UTC-5	2015 November 11 17:56:40 UTC-5
Successful	Launching a new EC2 instance: i-9e05a12e	2015 November 11 17:26:25 UTC-5	2015 November 11 17:27:01 UTC-5
Successful	Launching a new EC2 instance: i-9f05a12f	2015 November 11 17:26:25 UTC-5	2015 November 11 17:26:59 UTC-5
Successful	Launching a new EC2 instance: i-30367387	2015 November 11 17:26:24 UTC-5	2015 November 11 17:27:27 UTC-5
Successful	Launching a new EC2 instance: i-31367386	2015 November 11 17:26:24 UTC-5	2015 November 11 17:27:29 UTC-5

Auto Scaling Group: my_asg

Instances

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	2	2	2	10	us-east-1c, us-east-1b	300	100

Auto Scaling Group: my_asg

Instances

Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status
i-31367386	InService	as_lc	us-east-1b	Healthy
i-9e05a12e	InService	as_lc	us-east-1c	Healthy

The load balancer is adjusted dynamically.

Filter: X

Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count
myLB	myLB-221011144.us-east-1....	80 (HTTP) forwarding to 80 (...)	us-east-1a, us-east-1b,...	2 Instances

Load balancer: myLB

Description Instances Health Check Monitoring Security Listeners Tags

DNS Name: myLB-221011144.us-east-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see [Using Domain Names With Elastic Load Balancing](#).

Scheme: internet-facing

Status: 2 of 2 instances in service

Load balancer: myLB

Description Instances Health Check Monitoring Security Listeners Tags

Connection Draining: Enabled, 300 seconds ([Edit](#))

[Edit Instances](#)

Instance ID	Name	Availability Zone	Status	Actions
i-9e05a12e		us-east-1c	InService ⓘ	Remove from Load Balancer
i-31367386		us-east-1b	InService ⓘ	Remove from Load Balancer

Below we can see when the alarm was fired, it sent notifications to the subscribe and trigger the scale down policy. When the “ApproximateNumberOfMessagesVisible” was above 50, this alarm would change to “OK”.

Filter: State is OK X

State	Name	Threshold	Config Status
<input checked="" type="checkbox"/> OK	DownwardMovingSQSAlarm	ApproximateNumberOfMessagesVisible <= 50 for 2 minutes	
<input type="checkbox"/> OK	UpwardMovingSQSAlarm	ApproximateNumberOfMessagesVisible >= 200 for 2 minutes	

1 Alarm selected

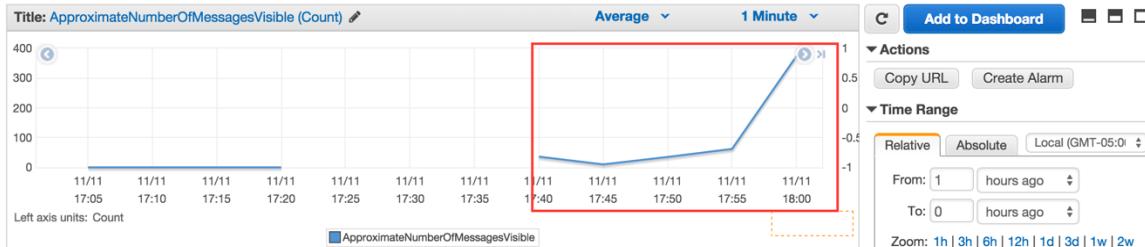
Alarm: DownwardMovingSQSAlarm

Details History

Showing all history entries (5)

Date	Type	Description
2015-11-11 17:56 UTC-5	State update	Alarm updated from ALARM to OK
2015-11-11 17:54 UTC-5	Action	Successfully executed action arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:edff66d5-94ef-4519-b079-4259e...
2015-11-11 17:54 UTC-5	Action	Successfully executed action arn:aws:sns:us-east-1:217134905396:ASTopic
2015-11-11 17:54 UTC-5	State update	Alarm updated from INSUFFICIENT_DATA to ALARM
2015-11-11 17:54 UTC-5	Configuration update	Alarm "DownwardMovingSQSAlarm" created

When the “ApproximateNumberOfMessagesVisible” in the queue was more than 200. The “UpwardingSQSAlarm” was fired.



CloudWatch Alarms page showing the configuration for the 'UpwardMovingSQSAlarm'. The alarm state changed to ALARM at 2015/11/11, with one datapoint (374.0) crossing the threshold (200.0). The actions include scaling up the Auto Scaling group 'my_asg' and sending a message to the SNS topic 'ASTopic'. A linked CloudWatch Metrics chart shows the metric 'ApproximateNumberOfMessagesVisible' rising from 200 to 374 at the same time.

The alarm will notify the subscriber and trigger the scale up policy and add 3 instances.

Email notification from AWS SNS to the 'ASTopic' subscriber. The email body contains the following information:

- Subject:** ALARM: "UpwardMovingSQSAlarm" in US - N. Virginia
- From:** ASTopic <no-reply@sns.amazonaws.com>
- To:** me
- Timestamp:** 6:03 PM (1 minute ago)
- Message Content:**

You are receiving this email because your Amazon CloudWatch Alarm "UpwardMovingSQSAlarm" in the US - N. Virginia region has entered the ALARM state, because "Threshold Crossed: 1 datapoint (374.0) was greater than or equal to the threshold (200.0)." at "Wednesday 11 November, 2015 23:03:54 UTC".

View this alarm in the AWS Management Console:
<https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#s=Alarms&alarm=UpwardMovingSQSAlarm>

Alarm Details:

 - Name: UpwardMovingSQSAlarm
 - Description: UpwardMovingSQSAlarm
 - State Change: OK -> ALARM
 - Reason for State Change: Threshold Crossed: 1 datapoint (374.0) was greater than or equal to the threshold (200.0).
 - Timestamp: Wednesday 11 November, 2015 23:03:54 UTC
 - AWS Account: 217134905396

Threshold:

 - The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 200.0 for 60 seconds.

Monitored Metric:

 - MetricNamespace: AWS/SQS
 - MetricName: ApproximateNumberOfMessagesVisible
 - Dimensions: [QueueName = MyQueue]
 - Period: 60 seconds
 - Statistic: Average
 - Unit: not specified

State Change Actions:

 - OK:
 - ALARM: [arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ced14c30-e0b0-46f8-8c45-5f8214b6a210:autoScalingGroupName/my_asg:policyName/scale-up] [arn:aws:sns:us-east-1:217134905396:ASTopic]
 - INSUFFICIENT_DATA:

Launch 3 new instances. The desired number of machines will become 5 after refresh.

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	2	2	2	10	us-east-1c, us-east-1b	300	100

Auto Scaling Group: my_asg

Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status ▾			Filter scaling history...	1 to 9 of 9 History Items		
Status	Description	Start Time	End Time			
▶ Successful	Launching a new EC2 instance: i-f17c3946	2015 November 11 18:04:24 UTC-5	2015 November 11 18:04:58 UTC-5			
▶ Successful	Launching a new EC2 instance: i-a1f55211	2015 November 11 18:04:24 UTC-5	2015 November 11 18:04:59 UTC-5			
▶ Successful	Launching a new EC2 instance: i-a0f55210	2015 November 11 18:04:24 UTC-5	2015 November 11 18:04:58 UTC-5			
▶ Successful	Terminating EC2 instance: i-9f05a12f	2015 November 11 17:54:51 UTC-5	2015 November 11 17:56:07 UTC-5			
▶ Successful	Terminating EC2 instance: i-30367387	2015 November 11 17:54:50 UTC-5	2015 November 11 17:56:40 UTC-5			

The alarm would only be fired once. If the “ApproximateNumberOfMessagesVisible” was still above 200 during the next check, the alarm would not be fired a second time. But the scale up policy would be triggered again. So the “Desired” number of instances changed to 8. It would keep increasing until it reached the “Max” value if the alarm was still in the “ALARM” state.

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	8	8	2	10	us-east-1c, us-east-1b	300	100

Auto Scaling Group: my_asg

Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status ▾			Filter scaling history...	1 to 12 of 12 History Items		
Status	Description	Start Time	End Time			
▶ Successful	Launching a new EC2 instance: i-68f651d8	2015 November 11 18:06:25 UTC-5	2015 November 11 18:07:30 UTC-5			
▶ Successful	Launching a new EC2 instance: i-0b783dbc	2015 November 11 18:06:25 UTC-5	2015 November 11 18:07:31 UTC-5			
▶ Successful	Launching a new EC2 instance: i-3783dc4	2015 November 11 18:06:25 UTC-5	2015 November 11 18:07:01 UTC-5			
▶ Successful	Launching a new EC2 instance: i-f17c3946	2015 November 11 18:04:24 UTC-5	2015 November 11 18:04:58 UTC-5			

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	10	10	2	10	us-east-1c, us-east-1b	300	100

Auto Scaling Group: my_asg

Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status ▾			Filter scaling history...	1 to 14 of 14 History Items		
Status	Description	Start Time	End Time			
▶ Successful	Launching a new EC2 instance: i-8984c13e	2015 November 11 18:09:26 UTC-5	2015 November 11 18:10:01 UTC-5			
▶ Successful	Launching a new EC2 instance: i-1af057aa	2015 November 11 18:09:25 UTC-5	2015 November 11 18:09:59 UTC-5			
▶ Successful	Launching a new EC2 instance: i-68f651d8	2015 November 11 18:06:25 UTC-5	2015 November 11 18:07:30 UTC-5			
▶ Successful	Launching a new EC2 instance: i-0b783dbc	2015 November 11 18:06:25 UTC-5	2015 November 11 18:07:31 UTC-5			

The load balancer would change accordingly.

Filter: Search Load Balancers

Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count
myLB	myLB-221011144.us-east-1....	80 (HTTP) forwarding to 80 (HTTP)	us-east-1a, us-east-1b,...	10 Instances

Load balancer: myLB

Description **Instances** **Health Check** **Monitoring** **Security** **Listeners** **Tags**

DNS Name: myLB-221011144.us-east-1.elb.amazonaws.com (A Record)

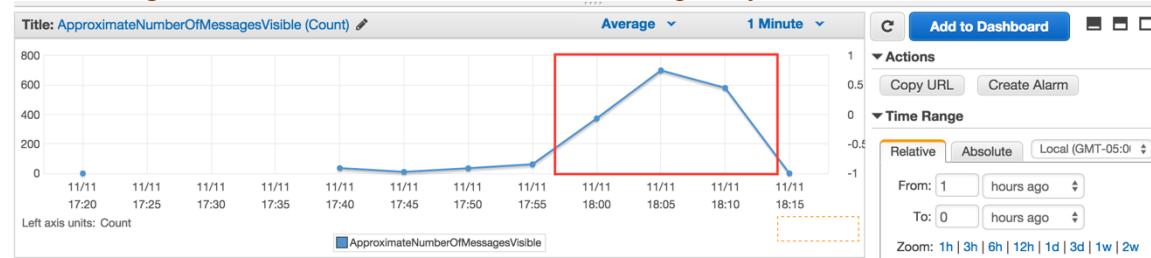
Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see [Using Domain Names With Elastic Load Balancing](#).

Scheme: internet-facing

Status: **9 of 10 instances in service**

Port Configuration: 80 (HTTP) forwarding to 80 (HTTP)
Stickiness: Disabled [\(Edit\)](#)

The messages in the SQS Queue will be consumed quickly.



When the “ApproximateNumberOfMessagesVisible” went below 200, the “DownwardMovingSQSAlarm” would change to “OK” state.

Alarms

State	Name	Threshold	Config Status
OK	UpwardMovingSQSAlarm	ApproximateNumberOfMessagesVisible >= 200 for 2 minutes	OK
OK	DownwardMovingSQSAlarm	ApproximateNumberOfMessagesVisible <= 50 for 2 minutes	OK

1 Alarm selected

Alarm: DownwardMovingSQSAlarm

Details **History**

State Details: State changed to OK at 2015/11/11. Reason: Threshold Crossed: 1 datapoint (62.0) was not less than or equal to the threshold (50.0).

Description: DownwardMovingSQSAlarm

Threshold: ApproximateNumberOfMessagesVisible <= 50 for 2 minutes

Actions: In ALARM: For group **my_asg** use policy **scale-dn** (Remove 2 instances)
Send message to topic "ASTopic" (hanjiaoju@g.harvard.edu)

Namespace: AWS/SQS
Metric Name: ApproximateNumberOfMessagesVisible
Dimensions: QueueName = MyQueue

DownwardMovingSQSAlarm ApproximateNumberOfMessagesVisible <= 50

6) I repeated the procedure the second time and run it much longer this time.

Like before, the “DownwardMovingSQSAlarm” would be alarmed first and decreased the desired instances to 2. When the number of message went above 200, the “UpwardMovingSQSAlarm” got fired and increased the desired instances to 5.

Filter: Filter Auto Scaling groups... X

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	5	5	2	10	us-east-1c, us-east-1b	100	100

Auto Scaling Group: my_asg

Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status Filter scaling history... X

Status	Description	Start Time	End Time
Successful	Launching a new EC2 instance: i-c8ea4778	2015 November 12 17:09:21 UTC-5	2015 November 12 17:09:59 UTC-5
Successful	Launching a new EC2 instance: i-a2d29f15	2015 November 12 17:09:20 UTC-5	2015 November 12 17:09:59 UTC-5
Successful	Launching a new EC2 instance: i-a1d29f16	2015 November 12 17:09:20 UTC-5	2015 November 12 17:10:04 UTC-5

Description: Launching a new EC2 instance: i-a1d29f16

Cause: At 2015-11-12T22:08:55Z a monitor alarm UpwardMovingSQSAlarm in state ALARM triggered policy scale-up changing the desired capacity from 2 to 5. At 2015-11-12T22:09:18Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 5.

When we had 10 instances running, the messages were consumed very fast. The number fell below the downward threshold. The “DownwardMovingSQSAlarm” would fire again and decrease the desired instances to 8, and then 6 if still too many instances.

Filter: Filter Auto Scaling groups... X

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
my_asg	as_lc	8	6	2	10	us-east-1c, us-east-1b	100	100

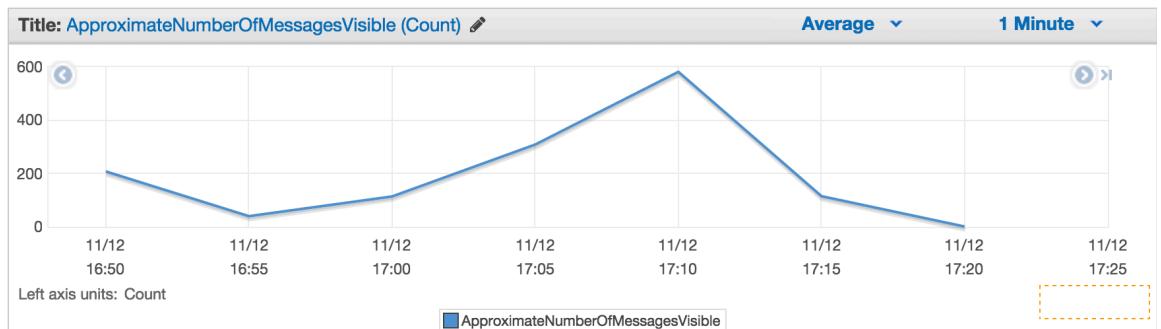
Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status Filter scaling history... X

Status	Description	Start Time	End Time
Waiting for ELB connection draining	Terminating EC2 instance: i-c8ea4778 - Waiting For...	2015 November 12 17:23:22 UTC-5	
Waiting for ELB connection draining	Terminating EC2 instance: i-a2d29f15 - Waiting For...	2015 November 12 17:23:22 UTC-5	
Successful	Terminating EC2 instance: i-d1c88566	2015 November 12 17:21:51 UTC-5	2015 November 12 17:23:12 UTC-5
Successful	Terminating EC2 instance: i-370daf87	2015 November 12 17:21:51 UTC-5	2015 November 12 17:23:02 UTC-5

Description: Terminating EC2 instance: i-370daf87

Cause: At 2015-11-12T22:21:20Z a monitor alarm DownwardMovingSQSAlarm in state ALARM triggered policy scale-in changing the desired capacity from 10 to 8. At 2015-11-12T22:21:51Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 10 to 8. At 2015-11-12T22:21:51Z instance i-370daf87 was selected for termination. At 2015-11-12T22:21:51Z instance i-d1c88566 was selected for termination.



Total Points: 30

Problem 2:

Please visit Auto Scaling documentation pages and find a way to create a fixed schedule for an auto scaling group. Let your group have 3 instances for one particular hour during the day and then 1 instance for the rest of the day. You do not have to run your test 24 hours to prove that scheduler works. Run it for three hours.

- 1) Create fixed schedule to set “desired-capacity” to 3 every 10 mins. Run it for 1 hr.

```
hqiu@bos-mpdei>> aws autoscaling put-scheduled-update-group-action --  
auto-scaling-group-name my_asg --scheduled-action-name scheduled_as_1 -  
--start-time "2015-11-12T06:00:00Z" --end-time "2015-11-12T07:00:00Z" --  
recurrence "*/10 * * * *" --min-size 2 --max-size 10 --desired-capacity  
3
```

- 2) Create a fixed schedule to set “desired-capacity” to 1 every 10 mins. Run it for 2 hours after the first schedule had finished.

```
hqiu@bos-mpdei>> aws autoscaling put-scheduled-update-group-action --  
auto-scaling-group-name my_asg --scheduled-action-name scheduled_as_2 -  
--start-time "2015-11-12T07:02:00Z" --end-time "2015-11-12T09:02:00Z" --  
recurrence "*/10 * * * *" --min-size 1 --max-size 5 --desired-capacity  
1
```

- 3) Create a fixed schedule to set “desired-capacity” to 1 every 20 mins. Run it for 1 hour and postpone it one hour after the second schedule had finished.

```
hqiu@bos-mpdei>> aws autoscaling put-scheduled-update-group-action --  
auto-scaling-group-name my_asg --scheduled-action-name scheduled_as_3 -  
--start-time "2015-11-12T10:05:00Z" --end-time "2015-11-12T11:05:00Z" --  
recurrence "*/20 * * * *" --min-size 1 --max-size 6 --desired-capacity  
1
```

```
hqiu@bos-mpdei>> aws autoscaling describe-scheduled-actions  
{  
    "ScheduledUpdateGroupActions": [  
        {  
            "MinSize": 2,  
            "DesiredCapacity": 3,  
            "AutoScalingGroupName": "my_asg",  
            "MaxSize": 10,  
            "Recurrence": "*/10 * * * *"  
            "ScheduledActionARN": "arn:aws:autoscaling:us-east-  
1:217134905396:scheduledUpdateGroupAction:649ac5ce-c037-47c2-8179-  
4ffc67bb5207:autoScalingGroupName/my_asg:scheduledActionName/scheduled_  
as_1",  
            "ScheduledActionName": "scheduled_as_1",  
            "StartTime": "2015-11-12T06:00:00Z",  
            "Time": "2015-11-12T06:00:00Z",  
            "EndTime": "2015-11-12T07:00:00Z"  
        },  
        {  
            "MinSize": 1,
```

```

        "DesiredCapacity": 1,
        "AutoScalingGroupName": "my_asg",
        "MaxSize": 5,
        "Recurrence": "*/*/*/*/*",
        "ScheduledActionARN": "arn:aws:autoscaling:us-east-
1:217134905396:scheduledUpdateGroupAction:817c2c3d-971b-4c6b-989d-
ca14b45b816f:autoScalingGroupName/my_asg:scheduledActionName/scheduled_
as_2",
        "ScheduledActionName": "scheduled_as_2",
        "StartTime": "2015-11-12T07:02:00Z",
        "Time": "2015-11-12T07:02:00Z",
        "EndTime": "2015-11-12T09:02:00Z"
    },
    {
        "MinSize": 1,
        "DesiredCapacity": 1,
        "AutoScalingGroupName": "my_asg",
        "MaxSize": 6,
        "Recurrence": "*/*/*/*/*",
        "ScheduledActionARN": "arn:aws:autoscaling:us-east-
1:217134905396:scheduledUpdateGroupAction:3512ae13-add7-43b9-844e-
087b0b8eb8ba:autoScalingGroupName/my_asg:scheduledActionName/scheduled_
as_3",
        "ScheduledActionName": "scheduled_as_3",
        "StartTime": "2015-11-12T10:05:00Z",
        "Time": "2015-11-12T10:05:00Z",
        "EndTime": "2015-11-12T11:05:00Z"
    }
]
}

```

4) Results after the first scheduled had been launched (“Desired” = 3).

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
my_asg	as_lc	9	3	2	10	us-east-1c, us-east-1b	100	10

Auto Scaling Group: my_asg

Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status ▾ Filter scaling history... X

K < 1 to 25 of 103 History Items > |

Status	Description	Start Time	End Time
Successful	Executing scheduled action scheduled_as_1	2015 November 12 01:00:04 UTC-5	2015 November 12 01:00:04 UTC-5
	Description: Executing scheduled action scheduled_as_1		
	Cause:		

Status	Description	Start Time	End Time
In progress	Terminating EC2 instance: i-ac2a8c1c	2015 November 12 01:00:35 UTC-5	
In progress	Terminating EC2 instance: i-bdfcbb0a	2015 November 12 01:00:35 UTC-5	
In progress	Terminating EC2 instance: i-ab2a8c1b	2015 November 12 01:00:35 UTC-5	
In progress	Terminating EC2 instance: i-682e88d8	2015 November 12 01:00:34 UTC-5	
In progress	Terminating EC2 instance: i-21f3b496	2015 November 12 01:00:34 UTC-5	

Description: Terminating EC2 instance: i-21f3b496

Cause: At 2015-11-12T06:00:04Z a scheduled action update of AutoScalingGroup constraints to min: 2, max: 10, desired: 3 changing the desired capacity from 8 to 3. At 2015-11-12T06:00:04Z the scheduled action scheduled_as_1 executed. Setting desired capacity from 8 to 3. At 2015-11-12T06:00:34Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 8 to 3. At 2015-11-12T06:00:34Z instance i-21f3b496 was selected for termination. At 2015-11-12T06:00:34Z instance i-682e88d8 was selected for termination. At 2015-11-12T06:00:35Z instance i-ab2a8c1b was selected for termination. At 2015-11-12T06:00:35Z instance i-bdfcbb0a was selected for termination. At 2015-11-12T06:00:35Z instance i-ac2a8c1c was selected for termination.

Since I manually increase the speed to push messages into the queue, the messages got piled up again. A “DownScalingSQSAlarm” was fired and decreased the “desired-capacity” to 2.

Status	Description	Start Time	End Time
Waiting for ELB connection draining	Terminating EC2 instance: i-23fabd94 - Waiting For...	2015 November 12 01:03:06 UTC-5	

Description: Terminating EC2 instance: i-23fabd94 - Waiting For ELB Connection Draining.

Cause: At 2015-11-12T06:02:37Z a monitor alarm DownScalingSQSAlarm in state ALARM triggered policy scale-dn changing the desired capacity from 3 to 2. At 2015-11-12T06:03:06Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 3 to 2. At 2015-11-12T06:03:06Z instance i-23fabd94 was selected for termination.

The “scheduled_as_1” was launched every 10 mins and set the “Desired” to 3.

Status	Description	Start Time	End Time
Successful	Launching a new EC2 instance: i-3f266088	2015 November 12 01:10:38 UTC-5	2015 November 12 01:11:12 UTC-5
	Description: Launching a new EC2 instance: i-3f266088		
	Cause: At 2015-11-12T06:10:06Z a scheduled action update of AutoScalingGroup constraints to min: 2, max: 10, desired: 3 changing the desired capacity from 2 to 3. At 2015-11-12T06:10:06Z the scheduled action scheduled_as_1 executed. Setting desired capacity from 2 to 3. At 2015-11-12T06:10:36Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3.		
Successful	Executing scheduled action scheduled_as_1	2015 November 12 01:10:06 UTC-5	2015 November 12 01:10:06 UTC-5
Successful	Terminating EC2 instance: i-23fabd94	2015 November 12 01:03:06 UTC-5	2015 November 12 01:04:32 UTC-5
Successful	Terminating EC2 instance: i-ac2a8c1c	2015 November 12 01:00:35 UTC-5	2015 November 12 01:01:43 UTC-5
Successful	Terminating EC2 instance: i-bdfcbb0a	2015 November 12 01:00:35 UTC-5	2015 November 12 01:01:35 UTC-5
Successful	Terminating EC2 instance: i-ab2a8c1b	2015 November 12 01:00:35 UTC-5	2015 November 12 01:01:45 UTC-5
Successful	Terminating EC2 instance: i-682e88d8	2015 November 12 01:00:34 UTC-5	2015 November 12 01:01:37 UTC-5
Successful	Terminating EC2 instance: i-21f3b496	2015 November 12 01:00:34 UTC-5	2015 November 12 01:01:55 UTC-5
Successful	Executing scheduled action scheduled_as_1	2015 November 12 01:00:04 UTC-5	2015 November 12 01:00:04 UTC-5
Successful	Terminating EC2 instance: i-b7eaad00	2015 November 12 00:59:04 UTC-5	2015 November 12 01:00:29 UTC-5

5) Results after the second and third schedule had been launched.

▼ Successful	Terminating EC2 instance: i-77d395c0	2015 November 12 03:31:00 UTC-5	2015 November 12 03:32:30 UTC-5
Description: Terminating EC2 instance: i-77d395c0			
▶ Successful	Executing scheduled action scheduled_as_2	2015 November 12 03:30:30 UTC-5	2015 November 12 03:30:30 UTC-5

The “scheduled_as_2” was launched every 10 mins. The auto scaling group will work as usual. For example, here it terminated 3 instances due to executing “schedule_as_2”. But since the number of messages in the queue is above 200. The “UpScalingSQSAlarm” would be fired or it was already in the “ALARM” state.

The “UpScalingSQSAlarm” would launch 3 instances afterwards until it reached the “Max” value.

▶ Successful	Executing scheduled action scheduled_as_2	2015 November 12 03:40:03 UTC-5	2015 November 12 03:40:03 UTC-5
▶ Successful	Launching a new EC2 instance: i-538b2ae3	2015 November 12 03:39:14 UTC-5	2015 November 12 03:39:51 UTC-5
▶ Successful	Terminating EC2 instance: i-44f1b7f3	2015 November 12 03:38:33 UTC-5	2015 November 12 03:39:50 UTC-5
▶ Successful	Launching a new EC2 instance: i-44f1b7f3	2015 November 12 03:36:34 UTC-5	2015 November 12 03:37:07 UTC-5
▶ Successful	Launching a new EC2 instance: i-a842512	2015 November 12 03:33:33 UTC-5	2015 November 12 03:34:06 UTC-5
▶ Successful	Launching a new EC2 instance: i-89f5b33f	2015 November 12 03:33:33 UTC-5	2015 November 12 03:34:36 UTC-5
▶ Successful	Launching a new EC2 instance: i-89f5b33e	2015 November 12 03:33:33 UTC-5	2015 November 12 03:34:07 UTC-5
▶ Successful	Terminating EC2 instance: i-eae5a35d	2015 November 12 03:31:00 UTC-5	2015 November 12 03:32:51 UTC-5
▶ Successful	Terminating EC2 instance: i-38bc1d88	2015 November 12 03:31:00 UTC-5	2015 November 12 03:32:08 UTC-5
▶ Successful	Terminating EC2 instance: i-77d395c0	2015 November 12 03:31:00 UTC-5	2015 November 12 03:32:30 UTC-5
▶ Successful	Executing scheduled action scheduled_as_2	2015 November 12 03:30:30 UTC-5	2015 November 12 03:30:30 UTC-5

The “scheduled_as_3” was executed.

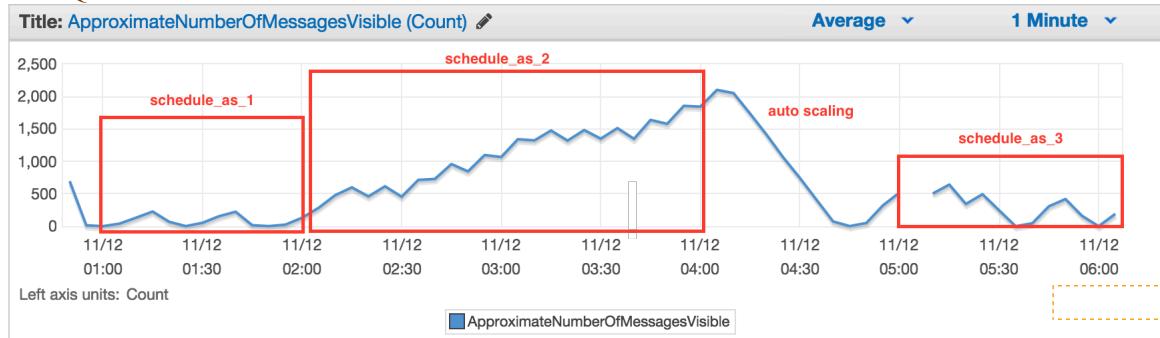
▼ Successful	Terminating EC2 instance: i-c3692874	2015 November 12 05:05:31 UTC-5	2015 November 12 05:06:30 UTC-5
Description: Terminating EC2 instance: i-c3692874			
▶ Successful	Executing scheduled action scheduled_as_3	2015 November 12 05:05:01 UTC-5	2015 November 12 05:05:01 UTC-5

The “scheduled_as_3” was launched every 20 mins.

▶ Successful	Executing scheduled action scheduled_as_3	2015 November 12 06:00:21 UTC-5	2015 November 12 06:00:21 UTC-5
▶ Successful	Launching a new EC2 instance: i-bdb4f50a	2015 November 12 05:51:23 UTC-5	2015 November 12 05:52:35 UTC-5
▶ Successful	Terminating EC2 instance: i-b4b3f203	2015 November 12 05:50:50 UTC-5	2015 November 12 05:51:56 UTC-5
▶ Successful	Launching a new EC2 instance: i-3203a282	2015 November 12 05:49:52 UTC-5	2015 November 12 05:50:57 UTC-5
▶ Successful	Launching a new EC2 instance: i-b4b3f203	2015 November 12 05:49:21 UTC-5	2015 November 12 05:49:54 UTC-5
▶ Successful	Launching a new EC2 instance: i-8a00a13a	2015 November 12 05:49:21 UTC-5	2015 November 12 05:50:25 UTC-5
▶ Successful	Terminating EC2 instance: i-aa3e9f1a	2015 November 12 05:49:15 UTC-5	2015 November 12 05:50:40 UTC-5
▶ Successful	Launching a new EC2 instance: i-c8afeef7	2015 November 12 05:46:16 UTC-5	2015 November 12 05:47:20 UTC-5
▶ Successful	Launching a new EC2 instance: i-a93e9f19	2015 November 12 05:46:16 UTC-5	2015 November 12 05:47:20 UTC-5
▶ Successful	Launching a new EC2 instance: i-aa3e9f1a	2015 November 12 05:46:16 UTC-5	2015 November 12 05:47:20 UTC-5
▶ Successful	Executing scheduled action scheduled_as_3	2015 November 12 05:40:14 UTC-5	2015 November 12 05:40:14 UTC-5

6) Analysis of the results.

The Queue status:



In the first stage, since the “Max” instances was 10 and the desired number of instances was 3. The auto scaling group would work very well. The number of messages was from 0 ~ 300.

In the second stage, since the “Max” instances was 5 and the desired number of instances was 1. The auto scaling group wouldn’t be able to consume all the messages. Everytime the “schedule_as_2” was executed, there was only 1 instance to consume the messages. The messages got piled up in the queue.

The “ScaleUpSQSAlarm” was always in the “ALARM” state during this period. It would keep adding instances after the cooldown time. Below we can see it changed back to “OK” until “04:42 UTC-5”.

State	Name	Threshold	Config Status
<input type="checkbox"/> ALARM	DownScalingSQSAlarm	ApproximateNumberOfMessagesVisible <= 50 for 1 minute	
<input checked="" type="checkbox"/> OK	UpScalingSQSAlarm	ApproximateNumberOfMessagesVisible >= 200 for 1 minute	

2015-11-12 05:11 UTC-5	Action	Successfully executed action arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ab3a1a10-cd46-4abf
2015-11-12 05:11 UTC-5	State update	Alarm updated from INSUFFICIENT_DATA to ALARM
2015-11-12 05:07 UTC-5	State update	Alarm updated from ALARM to INSUFFICIENT_DATA
2015-11-12 05:01 UTC-5	Action	Failed to execute action arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ab3a1a10-cd46-4abf-b325
2015-11-12 04:57 UTC-5	Action	Successfully executed action arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ab3a1a10-cd46-4abf
2015-11-12 04:57 UTC-5	State update	Alarm updated from OK to ALARM
2015-11-12 04:42 UTC-5	State update	Alarm updated from ALARM to OK
2015-11-12 02:07 UTC-5	Action	Successfully executed action arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ab3a1a10-cd46-4abf
2015-11-12 02:07 UTC-5	State update	Alarm updated from OK to ALARM
2015-11-12 01:47 UTC-5	State update	Alarm updated from ALARM to OK
2015-11-12 01:41 UTC-5	Action	Successfully executed action arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ab3a1a10-cd46-4abf
2015-11-12 01:41 UTC-5	State update	Alarm updated from OK to ALARM
2015-11-12 01:21 UTC-5	State update	Alarm updated from ALARM to OK
2015-11-12 01:16 UTC-5	Action	Successfully executed action arn:aws:autoscaling:us-east-1:217134905396:scalingPolicy:ab3a1a10-cd46-4abf

I postponed the “schedule_as_3” an hour after the second schedule finished. During this time, the auto scaling group did a good job and drew the number of messages back to a low number.

Since the “schedule_as_3” was executed every 20 mins and the “Max” instances was 6, it can still handle the messages and the performance was much better than the second stage.

Total Points: 30

Problem 3:

Once you are done with Problem 2, please try to terminate all of your instances. Please describe what is happening.

- 1) To make the results more clear, I first reset the “desired-capacity” to 4 and wait for all of the 4 instances up.

```
hqiu@bos-mpdei>> aws autoscaling update-auto-scaling-group --auto-scaling-group-name my_asg --min-size 2 --max-size 4 --desired-capacity 4
```

The screenshot shows the AWS Auto Scaling Groups console. At the top, there is a search bar labeled "Filter: Filter Auto Scaling groups...". Below the search bar, there is a table with the following columns: Name, Launch Configuration, Instances, Desired, Min, Max, Availability Zones, Default Cooldown, and Health Check Gr. A single row is selected, showing "my_asg" as the name, "as_lc" as the launch configuration, and 4 instances in all other fields. Below the table, the heading "Auto Scaling Group: my_asg" is displayed. Underneath, there are tabs for Details, Activity History, Scaling Policies, Instances (which is highlighted in orange), Notifications, and Tags. In the "Instances" section, there is a sub-table titled "Filter: Any Health Status ▾ Any Lifecycle State ▾ Filter instances...". This sub-table lists four instances: i-bdb4f50a, i-cb9d3d7b, i-cd64c47d, and i-de94d469, all in the "InService" lifecycle state and associated with the "as_lc" launch configuration. All four instances are listed as "Healthy".

Manually terminate all the instances in the EC2 console.

The screenshot shows the AWS EC2 Instances console. The table has the following columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Pub. There are four rows, each representing an instance: i-cb9d3d7b, i-bdb4f50a, i-de94d469, and i-cd64c47d. All four instances are in the "t2.micro" instance type, located in "us-east-1c" and "us-east-1b" availability zones. Their current state is "shutting-do...", indicated by a yellow circle icon. The status checks column shows "None" for all instances. The alarm status column shows a green bell icon for all instances. The public IP column is partially visible on the right.

The Load Balancer detected that the instances were out of service and notified the Auto Scaling Group. Since the “desired-capacity” is 4 now. The Auto Scaling Group will automatically terminate the out-reached instance and launch a new one. It will replace the terminated instances one by one until another 4 new instances were in service.

Status	Description	Start Time	End Time		
In progress	Terminating EC2 instance: i-cb9d3d7b	2015 November 12 08:31:01 UTC-5			
Description: Terminating EC2 instance: i-cb9d3d7b					
Cause: At 2015-11-12T13:31:01Z an instance was taken out of service in response to a EC2 health check indicating it has been terminated or stopped.					
Status	Description	Start Time	End Time		
Successful	Terminating EC2 instance: i-de94d469	2015 November 12 08:33:02 UTC-5			
Description: Terminating EC2 instance: i-de94d469					
Cause: At 2015-11-12T13:33:02Z an instance was taken out of service in response to a EC2 health check indicating it has been terminated or stopped.					
► Successful	Launching a new EC2 instance: i-4a68c8fa	2015 November 12 08:31:33 UTC-5			
► Successful	Terminating EC2 instance: i-cb9d3d7b	2015 November 12 08:31:01 UTC-5			
► Successful	Launching a new EC2 instance: i-8666c636	2015 November 12 08:29:33 UTC-5			
► Successful	Terminating EC2 instance: i-cd64c47d	2015 November 12 08:29:00 UTC-5			
► Successful	Launching a new EC2 instance: i-cd64c47d	2015 November 12 08:27:31 UTC-5			
 i-f471d144	t2.micro	us-east-1c	 pending	 Initializing	None
 i-6ca4e4db	t2.micro	us-east-1b	 running	 Initializing	None
 i-4fa8e8f8	t2.micro	us-east-1b	 running	 Initializing	None

If the Auto Scaling Group found an instance out of service for a period, it would do the same thing too, which terminate the instance and launch a new one.

The Auto Scaling Group will try to meet its “desired-capacity”. If we set it to 0, it will not create new instances even we terminate all the running instances.

In order to delete instances of an auto scaling group you first have to modify that group by reducing both its min-size and max-size to zero. Only then can you delete the group.

- 2) Delete all the instances in the auto scaling group and delete the group. Reduce the min-size, max-size and desired-capacity to 0. All the instances will be terminated automatically.

```
hqiu@bos-mpdei>> aws autoscaling update-auto-scaling-group --auto-scaling-group-name my_asg --min-size 0 --max-size 0 --desired-capacity 0
```

Status	Description	Start Time	End Time
► Successful	Terminating EC2 instance: i-f471d144	2015 November 12 08:39:06 UTC-5	2015 November 12 08:40:00 UTC-5
▼ Successful	Terminating EC2 instance: i-4fa8e8f8	2015 November 12 08:39:06 UTC-5	2015 November 12 08:40:15 UTC-5
Description: Terminating EC2 instance: i-4fa8e8f8			
Cause: At 2015-11-12T13:38:41Z a user request update of AutoScalingGroup constraints to min: 0, max: 0, desired: 0 changing the desired capacity from 4 to 0. At 2015-11-12T13:39:05Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 0. At 2015-11-12T13:39:06Z instance i-6ca4e4db was selected for termination. At 2015-11-12T13:39:06Z instance i-4fa8e8f8 was selected for termination. At 2015-11-12T13:39:06Z instance i-f471d144 was selected for termination.			
► Successful	Terminating EC2 instance: i-6ca4e4db	2015 November 12 08:39:06 UTC-5	2015 November 12 08:40:14 UTC-5
► Successful	Terminating EC2 instance: i-4a68c8fa	2015 November 12 08:39:05 UTC-5	2015 November 12 08:39:23 UTC-5

Delete the Auto Scaling Group:

```
hqiu@bos-mpdei>> aws autoscaling delete-auto-scaling-group --auto-scaling-group-name my_asg
```

Finally, please delete AMI-s, load balancers, auto scaling groups and launch configurations and other AWS resources you do not need any more.

Total Points: 15

Problem 4.

In your Eclipse create an AWS Lambda Project. Create a simple lambda function that will take a simple string as an input argument and write a duplicated string, e.g. stringstring to the output. Demonstrate that your lambda function works.

Total Points: 25

1) Create an IAM Administrative user. Sign into the AWS console as a new user.



First create an IAM Administrative group.

The screenshot shows the AWS IAM Groups page. It features a 'Create New Group' button and a 'Group Actions' dropdown. Below is a table with columns for Group Name, Users, Inline Policy, and Creation Time. A message at the bottom says 'No records found.'

Create New Group Wizard

Step 1: Group Name

Step 2: Attach Policy

Step 3: Review

Set Group Name

Specify a group name. Group names can be edited any time.

Group Name:

Administrator

Example: Developers or ProjectAlpha

Maximum 128 characters

Create New Group Wizard

Step 1: Group Name

Step 2: Attach Policy

Step 3: Review

Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Policy Name	Attached Entities	Creation Time
<input checked="" type="checkbox"/> AdministratorAccess	0	2015-02-06 13:39 EST
<input type="checkbox"/> AmazonAPIGatewayAdministrator	0	2015-07-09 13:34 EDT

Create New Group Wizard

Step 1: Group Name

Step 2: Attach Policy

Step 3: Review

Review

Review the following information, then click **Create Group** to proceed.

Group Name	Administrator
Policies	arn:aws:iam::aws:policy/AdministratorAccess

Create New Group **Group Actions ▾**

Group Name	Users	Inline Policy	Creation Time
<input type="checkbox"/> Administrator	0		2015-11-12 17:38 EST

Then create an IAM Administrative user.

Dashboard

Search IAM

Details

Groups

Users

Roles

Create New Users **User Actions ▾**

Enter User Names:

1. Admin

2.

3.

4.

5.

Maximum 64 characters each

Generate an access key for each user

Users need access keys to make secure REST or Query protocol requests to AWS service APIs.

For users who need access to the AWS Management Console, create a password in the Users panel after completing this wizard.

Add user to the group and set the password.

Create New Users **User Actions ▾**

User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
<input type="checkbox"/> Admin	0	N/A		1 active	2015-11-12 17:39 EST

Create New Users

User Actions ▾

- Add User to Groups**
- Delete User
- Manage Access Keys
- Manage Password
- Manage Signing Certificates
- Manage MFA Device
- Remove User from Groups

Select groups that user **Admin** will be added to.

Filter		Showing 1 of 1 item		
<input type="checkbox"/>	Group Name ▾	Users	Inline Policy	Creation Time ▾
<input checked="" type="checkbox"/>	Administrator	0		2015-11-12 17:38 EST

Create New Users

User Actions ▾

- Add User to Groups**
- Delete User
- Manage Access Keys**
- Manage Password**
- Manage Signing Certificates
- Manage MFA Device
- Remove User from Groups

Users who will be using the AWS Management Console require a password. Select from the options below to manage the password for user Admin.

Assign an auto-generated password

Assign a custom password

Password:

Confirm Password:

Visit <https://217134905396.signin.aws.amazon.com/console/> in browser.

Account:

User Name:

Password:

I have an MFA Token (more info)

Sign In

[Sign-in using root account credentials](#)

Internet of Things

AWS IoT BETA
Connect Devices to the cloud

Mobile Services

- Mobile Hub BETA**
Build, Test, and Monitor Mobile apps
- Cognito**
User Identity and App Data Synchronization
- Device Farm**
Test Android, Fire OS, and iOS apps on real devices in the Cloud
- Mobile Analytics**

Resource Groups

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Create a Group **Tag Editor**

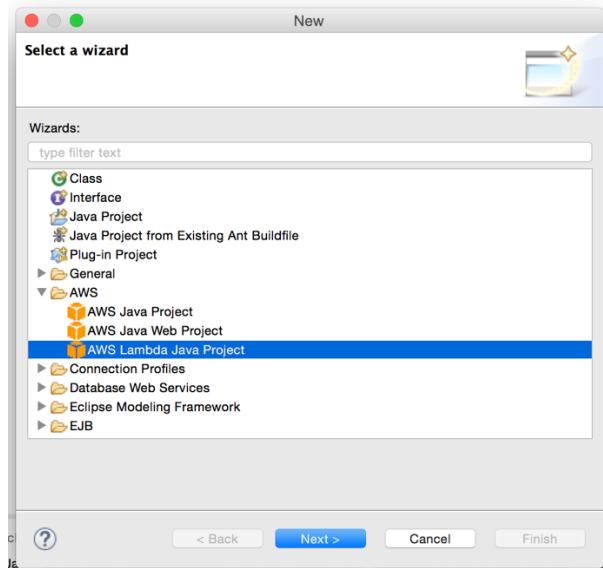
Additional Resources

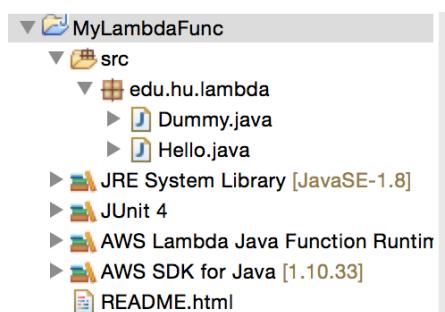
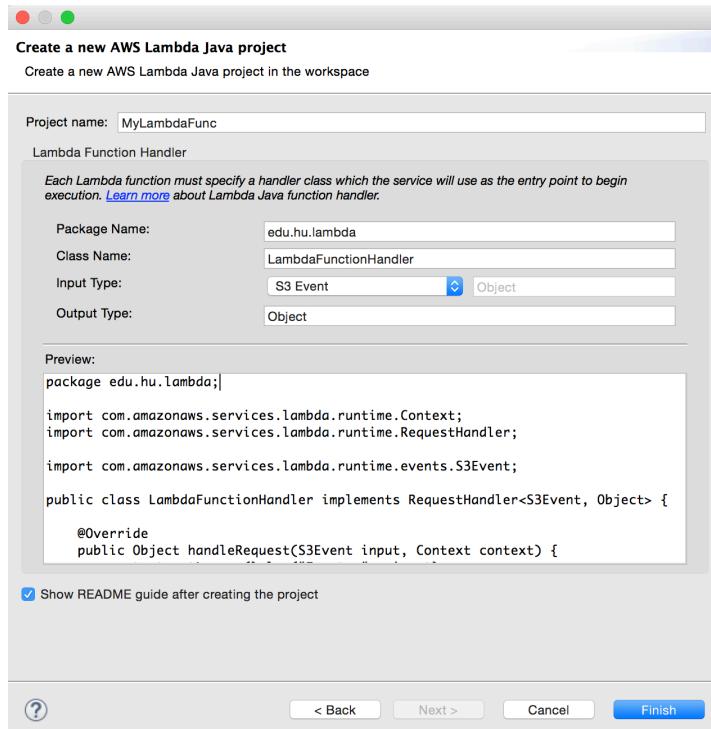
[Getting Started](#)

- 2) Create the AWS Lambda Project and run the lambda function. Here I tried two different ways to do it. The first method is way from professor's slides, which used "Hello.java" and "Dummy.java". The second way is following the link <http://docs.aws.amazon.com/lambda/latest/dg/java-handler-io-type-pojo.html>, which used "Hello2.java".

The first method ("Hello.java" and "Dummy.java"):

Create a AWS Lambda Java Project first.





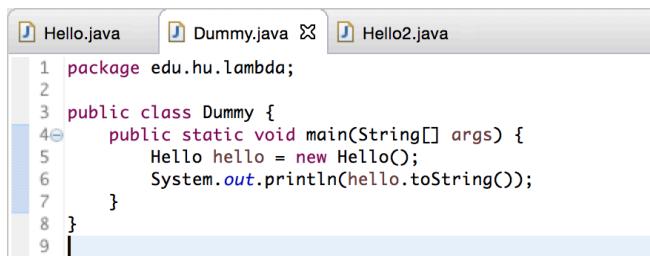
Hello.java:

```

1 package edu.hu.lambda;
2
3 import com.amazonaws.services.lambda.runtime.Context;
4 import com.amazonaws.services.lambda.runtime.LambdaLogger;
5
6 public class Hello {
7     public String myHandler(String myString, Context context) {
8         LambdaLogger logger = context.getLogger();
9         logger.log("received : " + myString);
10        return (myString + myString);
11    }
12 }
13

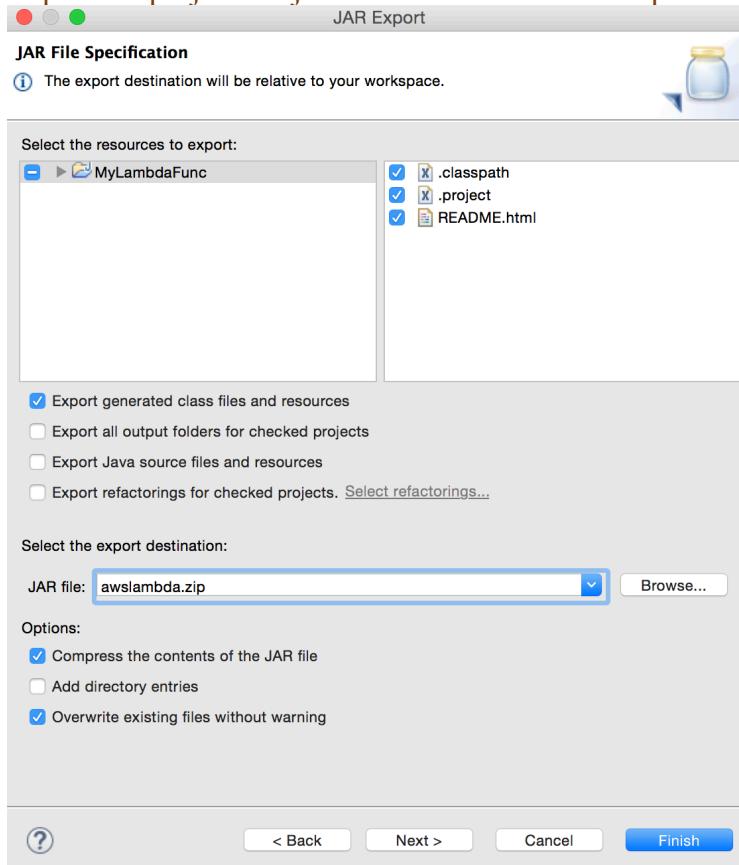
```

Dummy.java:



```
1 package edu.hu.lambda;
2
3 public class Dummy {
4     public static void main(String[] args) {
5         Hello hello = new Hello();
6         System.out.println(hello.toString());
7     }
8 }
9
```

Export the project to “jar” file and name it as a “zip” file.



- 3) Create and test lambda function using console.
Skip the blueprint page, configure the function. Upload the “awslambda.zip”.

Lambda > New function

Step 1: Select blueprint

Step 2: Configure function

Step 3: Review

Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name*

Description

Runtime*

Lambda function code

Code entry type Upload a .ZIP file Upload a .ZIP from Amazon S3

For .ZIP files larger than 10 MB, consider uploading via S3.

Lambda function handler and role

Handler*

Role*

Ensure that popups are enabled to create a new role. [Learn more](#) about Lambda execution roles.

Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. Changing memory or timeout may impact your function cost. [Learn more](#) about how Lambda pricing works.

Memory (MB)*

Timeout* min sec

For the Role, choose “lambda_basic_execution”.

AWS Lambda requires access to your resources

AWS Lambda uses an IAM role that grants your custom code permissions to access AWS resources it needs.

▼ Hide Details

Role Summary ?

Role Description Lambda execution role permissions

IAM Role

Role Name

▶ View Policy Document

The handler is “edu.hu.lambda.Hello::myHandler”.

Lambda > New function

Step 1: Select blueprint

Step 2: Configure function

Step 3: Review

Review

Please review your Lambda function details. You can go back to edit changes for each section. Complete the setup process.

Lambda function

Name hellolambda

Description Hello Lambda

Runtime Java 8

Handler `edu.hu.lambda.Hello::myHandler`

Role lambda_basic_execution

Memory (MB) 512

Timeout 15

Go to the function.

Lambda > Functions > hellolambda

ARN - arn:aws:lambda:us-east-1:217134905396:function:hellolambda

Save Save and test Actions ▾

Code Configuration Event sources API endpoints Monitoring ?

Runtime Java 8
Handler `edu.hu.lambda.Hello::myHandler` ⓘ
Role lambda_basic_execution ⓘ
Description Hello Lambda

Advanced settings

Select “Configure test event”.

Lambda > Functions > hellolambda

Save Save and test Actions ▾

Code Configuration Event sources API endpoints Monitoring ?

Configure test event
Publish new version
Create alias
Delete function
Download function code

Runtime Java 8
Handler `edu.hu.lambda.Hello::myHandler` ⓘ
Role lambda_basic_execution ⓘ
Description Hello Lambda

Advanced settings

Input test event

Use the editor below to enter an event to test your function with (please remember that

Sample event template Hello World

1 "hello"

Cancel Submit

It prints out the duplicated string.

Execution result: succeeded (logs)

"hellohello"

Summary		Log output	
Request ID	903522d1-8992-11e5-8658-2dfbdcb31303	The area below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.	
Duration	55.16 ms	START RequestId: 903522d1-8992-11e5-8658-2dfbdcb31303 Version: \$LATEST received : helloEND RequestId: 903522d1-8992-11e5-8658-2dfbdcb31303 REPORT RequestId: 903522d1-8992-11e5-8658-2dfbdcb31303 Duration: 55.16 ms Billed Duration: 100 ms Memory Size: 512 MB Max Memory Used: 66 MB	
Billed duration	100 ms		
Resources configured	512 MB		
Max memory used	66 MB		

Invoke the function from the command line:

```
hqiu@bos-mpdei>> aws lambda invoke --region us-east-1 --function-name hellolambda2 --payload 456 --invocation-type RequestResponse ~/Documents/workspace/response.txt
{
    "StatusCode": 200
}
```

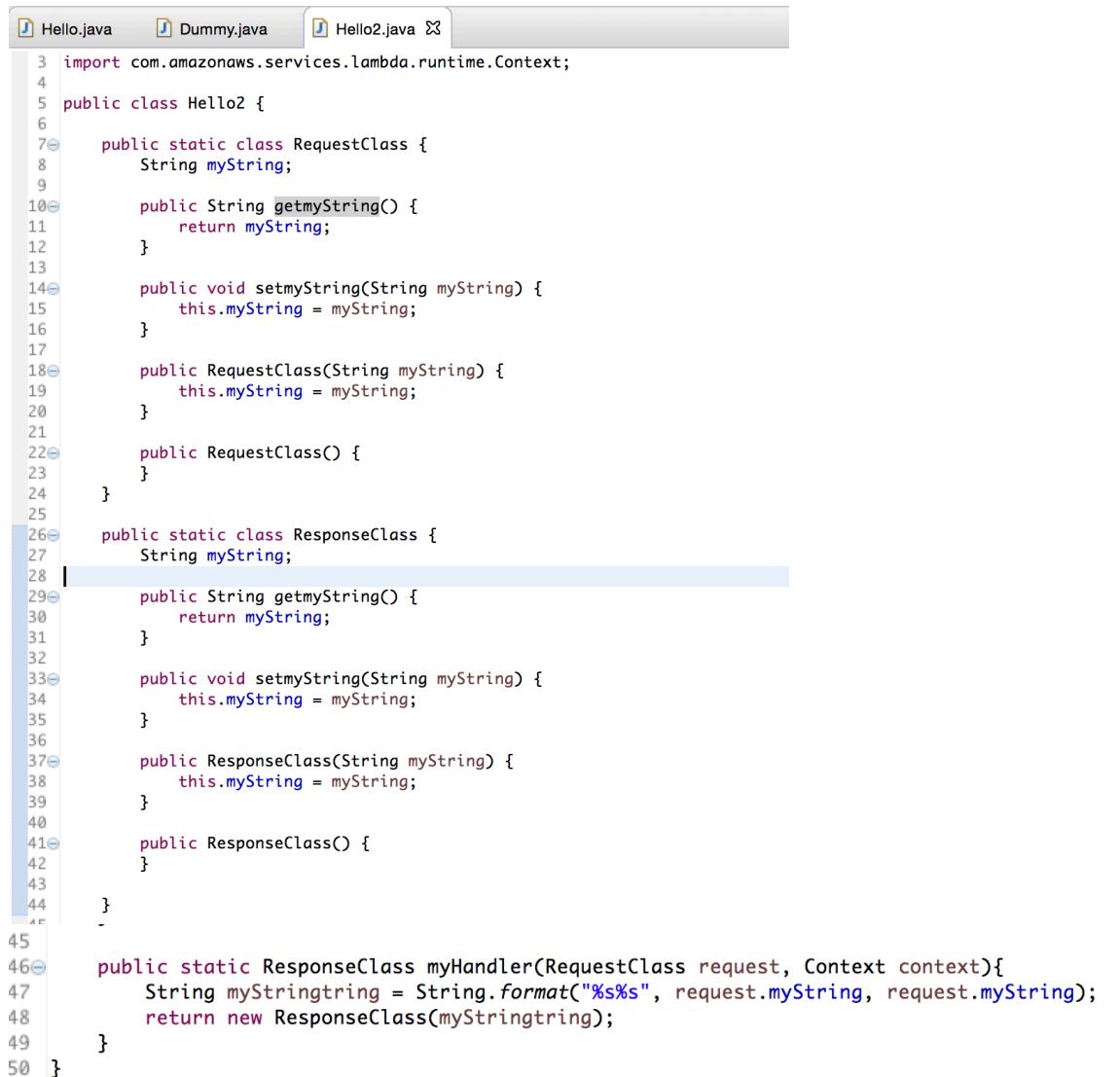
The “response.txt” will have “456456” in it.

- 4) The payload should be “JSON” format. I tried method 2 following the aws document. The “get” and “set” methods are required in order for the project to work with AWS Lambda's built in JSON serializer.

Second method (“Hello2.java”):

I created a RequestClass and a ResponseClass here.

Hello2.java:



```
3 import com.amazonaws.services.lambda.runtime.Context;
4
5 public class Hello2 {
6
7     public static class RequestClass {
8         String myString;
9
10    public String getmyString() {
11        return myString;
12    }
13
14    public void setmyString(String myString) {
15        this.myString = myString;
16    }
17
18    public RequestClass(String myString) {
19        this.myString = myString;
20    }
21
22    public RequestClass() {
23    }
24
25
26    public static class ResponseClass {
27        String myString;
28
29    public String getmyString() {
30        return myString;
31    }
32
33    public void setmyString(String myString) {
34        this.myString = myString;
35    }
36
37    public ResponseClass(String myString) {
38        this.myString = myString;
39    }
40
41    public ResponseClass() {
42    }
43
44    }
45
46    public static ResponseClass myHandler(RequestClass request, Context context){
47        String myStringstring = String.format("%s%s", request.myString, request.myString);
48        return new ResponseClass(myStringstring);
49    }
50 }
```

Generate the zip file. Create the lambda function through command line.

```
hqiu@bos-mpdei>> aws lambda create-function --region us-east-1 --
function-name hellolambda2 --zip-file
file:///~/Documents/workspace/awslambda2.zip --role
arn:aws:iam::217134905396:role/lambda_basic_execution --handler
edu.hu.lambda.Hello2::myHandler --runtime java8 --timeout 15 --memory-
size 512
{
    "FunctionName": "hellolambda2",
    "CodeSize": 4556,
    "MemorySize": 512,
    "FunctionArn": "arn:aws:lambda:us-east-
1:217134905396:function:hellolambda2",
    "Handler": "edu.hu.lambda.Hello2::myHandler",
    "Role": "arn:aws:iam::217134905396:role/lambda_basic_execution",
```

```

    "Timeout": 15,
    "LastModified": "2015-11-13T02:38:10.181+0000",
    "Runtime": "java8",
    "Description": ""
}

```

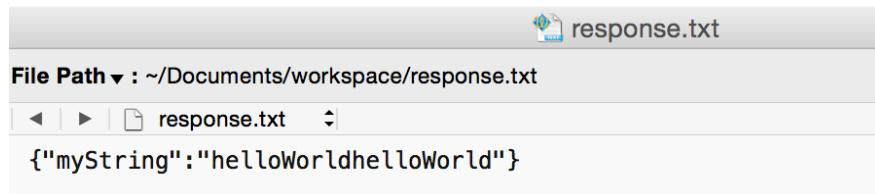
Invoke the lambda function. The payload is {"myString":"helloWorld"}.

```

hqiu@bos-mpdei>> aws lambda invoke --region us-east-1 --function-name
hellolambda2 --payload '{"myString":"helloWorld"}' --invocation-type
RequestResponse ~/Documents/workspace/response.txt
{
    "StatusCode": 200
}

```

Result file:



It still can be run from the console:

Input test event

Use the editor below to enter an event to test your function with (please remember that this will actually execute it).

Sample event template

```

1 {
2   "myString": "hello"
3 }

```

✓ Execution result: succeeded ([logs](#))

```

{
  "myString": "hellohello"
}

```

Summary

Request ID 50a3593e-89b0-11e5-b01a-b399a9882bd7

Duration 17.97 ms

Billed duration 100 ms

Resources configured 512 MB

Max memory used 30 MB