

HU Extension**Assignment 10
FINAL****E-90 Cloud Computing****Handed out: 11/07/2015****Due by 11:59 PM on Friday, 11/13/2015****Problem 1:**

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

Create an Auto Scaling launch configuration. Next create an auto scaling group. Create two autoscaling group policies, with respective names `scale-up` and `scale-dn.`, 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.

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

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.

Total Points: 30

For this solution several AWS components need to be created

1. A Linux instance with Apache, PHP, and Java installed with the `serversideconsumer.jar` running at boot. (HW9)
2. A custom AMI created from this Linux Instance (HW9)
3. An SQS queue for the messages that are produced and consumed with the Java apps
4. A SNS Topic that receives and transmits messages to subscribers in response to CloudWatch alarms
5. A Load Balancer
6. An Auto-scaling Group Launch Configuration
7. An Auto Scaling Group
8. An Auto-Scaling policy to scale up instances in response to too many messages in queue
9. An Auto-Scaling policy to scale down the number of instances when there is a low number of messages in the queue
10. A CloudWatch alarm configured to implement scale up policy when message number hits high threshold
11. A CloudWatch alarm configured to implement scale down policy when message number hits low threshold

Once all of these components are created and configured properly in relation to each other, messages can be produced with the Client producer app in Eclipse to test that CloudWatch triggers autoscaling up and down in response to message number.

Create a SQS queue named MyQueue

```
Marnie's-MacBook-Air:~ marnie$ aws sqs create-queue --queue-name MyQueue
{
    "QueueUrl": "https://queue.amazonaws.com/413513583861/MyQueue"
}
Marnie's-MacBook-Air:~ marnie$
```

Create a SNS Topic named AS_Test

```
Marnie's-MacBook-Air:~ marnie$ aws sns create-topic --name AS_Test
{
    "TopicArn": "arn:aws:sns:us-east-1:413513583861:AS_Test"
}
Marnie's-MacBook-Air:~ marnie$
```

Create a Load Balancer named LB

```
Marnie's-MacBook-Air:~ marnie$ aws elb create-load-balancer --load-balancer-name LB --listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" --subnets subnet-0bece57c --security-groups sg-52f9ed35
{
    "DNSName": "LB-1106736577.us-east-1.elb.amazonaws.com"
}
Marnie's-MacBook-Air:~ marnie$
```

Create Auto-scaling Group Launch Configuration

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling create-launch-configuration --launch-configuration-name lc --key-name ClKeyPair --image-id ami-2ec7bf44 --instance-type t1.micro --security-groups sg-52f9ed35
Marnie's-MacBook-Air:~ marnie$
```

Create Auto Scaling Group

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling create-auto-scaling-group --auto-scaling-group-name asg --launch-configuration-name lc --availability-zones us-east-1a us-east-1b --min-size 1 --max-size 10 --desired-capacity 4 --load-balancer-names LB --health-check-type ELB --health-check-grace-period 500
```

Create Auto-Scaling policy to scale up instances

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling put-scaling-policy --auto-scaling-group-name asg --policy-name scale-up --scaling-adjustment 1 --adjustment-type ChangeInCapacity --cool-down 50
{
    "PolicyARN": "arn:aws:autoscaling:us-east-1:413513583861:scalingPolicy:29b37789-c3d5-4aa1-8596-0573b94e6452:autoScalingGroupName/asg:policyName/scale-up"
}
Marnie's-MacBook-Air:~ marnie$
```

Create Auto-Scaling policy to scale down instances

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling put-scaling-policy --auto-scaling-group-name asg --policy-name scale-dn --scaling-adjustment -1 --adjustment-type ChangeInCapacity --cooldown 50
{
    "PolicyARN": "arn:aws:autoscaling:us-east-1:413513583861:scalingPolicy:138483e6-b8df-47b9-b5e9-e382f3f3e86:autoScalingGroupName/asg:policyName/scale-dn"
}
Marnie's-MacBook-Air:~ marnie$
```

Create CloudWatch alarm to scale up

1. Select Metric 2. Define Alarm

Name: <input type="text" value="scale_up"/>	Description: <input type="text" value="Add instances in response to message volume"/>						
Whenever: ApproximateNumberOfMessagesVisible							
is: <input type="button" value=">= 300"/>	for: <input type="text" value="1"/> consecutive period(s)						
Actions Define what actions are taken when your alarm changes state.							
Notification <table border="1"> <tr> <td>Whenever this alarm: <input type="button" value="State is ALARM"/></td> <td>Delete</td> </tr> <tr> <td>Send notification to: <input type="text" value="AS_Test"/> <input type="button" value="New list Enter list"/></td> <td><input type="button" value="Cancel"/></td> </tr> <tr> <td colspan="2">This notification list is managed in the SNS console.</td> </tr> </table>		Whenever this alarm: <input type="button" value="State is ALARM"/>	Delete	Send notification to: <input type="text" value="AS_Test"/> <input type="button" value="New list Enter list"/>	<input type="button" value="Cancel"/>	This notification list is managed in the SNS console.	
Whenever this alarm: <input type="button" value="State is ALARM"/>	Delete						
Send notification to: <input type="text" value="AS_Test"/> <input type="button" value="New list Enter list"/>	<input type="button" value="Cancel"/>						
This notification list is managed in the SNS console.							
AutoScaling Action <table border="1"> <tr> <td>Whenever this alarm: <input type="button" value="State is ALARM"/></td> <td>Delete</td> </tr> <tr> <td>From the group: <input type="text" value="asg"/></td> <td><input type="button" value="Cancel"/></td> </tr> <tr> <td>Take this action: <input type="button" value="scale-up - Add 1 instance"/></td> <td><input type="button" value="Save Changes"/></td> </tr> </table>		Whenever this alarm: <input type="button" value="State is ALARM"/>	Delete	From the group: <input type="text" value="asg"/>	<input type="button" value="Cancel"/>	Take this action: <input type="button" value="scale-up - Add 1 instance"/>	<input type="button" value="Save Changes"/>
Whenever this alarm: <input type="button" value="State is ALARM"/>	Delete						
From the group: <input type="text" value="asg"/>	<input type="button" value="Cancel"/>						
Take this action: <input type="button" value="scale-up - Add 1 instance"/>	<input type="button" value="Save Changes"/>						

Create CloudWatch alarm to scale down

Create Alarm

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

Alarm Threshold

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

Name:	scale_dn
Description:	Decrease the number of instances in response to

Whenever: ApproximateNumberOfMessagesVisible

is: \geq 100

for: 1 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification	Delete
Whenever this alarm: State is ALARM	

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

ApproximateNumberOfMessagesVisible ≥ 0

1
0.75
0.5
0.25
0

11/12 11/12 11/12
17:00 18:00 19:00

Namespace: AWS/SQS
QueueName: MyQueue
Metric Name: ApproximateNum-

Create Alarm

CloudWatch > Create Alarm

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

Actions

Define what actions are taken when your alarm changes state.

Notification	Delete
Whenever this alarm: State is ALARM	
Send notification to: AS_Test New list Enter list	
This notification list is managed in the SNS console.	

AutoScaling Action	Delete
Whenever this alarm: State is ALARM	
From the group: asg	
Take this action: scale-dn - Remove 1 instance	

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

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

Filter: State is OK		Search Alarms	1 to 2 of 2 Alar
State	Name	Threshold	Config Status
OK	scale_dn	ApproximateNumberOfMessagesVisible >= 100 for 5 minutes	No notifications
OK	scale_up	ApproximateNumberOfMessagesVisible >= 300 for 1 minute	No notifications

5 of 25

Instances before messages

(First Instance is what AMI is based on. Other instances created)

In response to Autoscaling desired capacity of 4 , 4 instances were created. Since there are no messages in the queue the instances were then terminated according to the scale-dn policy)

Filter by tags and attributes or search by keyword							
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	First Instance	i-9f035628	t1.micro	us-east-1b	running	2/2 checks passed	None
		i-5ace60e4	t1.micro	us-east-1a	running	2/2 checks passed	None
		i-b0bfea07	t1.micro	us-east-1b	shutting-down		None
		i-5bce60e5	t1.micro	us-east-1a	terminated		None
		i-b1bfea06	t1.micro	us-east-1b	terminated		None

Only one instance in Auto scaling group and LB before creating messages

Filter: Filter Auto Scaling groups...							
	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones
	asg	lc	1	1	1	10	us-east-1b, us-east-1a

Auto Scaling Group: asg

Details	Activity History	Scaling Policies	Instances	Notifications	Tags
Actions ▾					
Filter: Any Health Status ▾ Any Lifecycle State ▾ Filter instances...					
Instance ID	Lifecycle	Launch Configuration Name	Availability Zone		
i-5ace60e4	InService	lc	us-east-1a		

Filter: X

	Load Balancer Name	DNS Name	Port Config
<input type="checkbox"/>	LB	LB-1106736577.us-east-1.el...	80 (HTTP)

Load balancer: LB

Description Instances Health Check Monitoring Security

Connection Draining: Disabled (Edit)

Edit Instances

Instance ID	Name	Availability Zone
i-5ace60e4		us-east-1a

Create Messages from Client to trigger scaling

Console Servers
ClientSideProducer [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk

PRODUCER

Sending a message: Message for assign10 at: 2015/11/13 16:07:20
Sending a message: Message for assign10 at: 2015/11/13 16:07:22
Sending a message: Message for assign10 at: 2015/11/13 16:07:23
Sending a message: Message for assign10 at: 2015/11/13 16:07:24
Sending a message: Message for assign10 at: 2015/11/13 16:07:26
Sending a message: Message for assign10 at: 2015/11/13 16:07:27
Sending a message: Message for assign10 at: 2015/11/13 16:07:28
Sending a message: Message for assign10 at: 2015/11/13 16:07:29
Sending a message: Message for assign10 at: 2015/11/13 16:07:30

Large number of messages in queue

The screenshot shows the AWS Lambda Queue console. At the top, there are buttons for 'Create New Queue' and 'Queue Actions'. A search bar labeled 'Filter by Prefix:' is present. Below the header is a table with columns: Name, Messages Available, Messages in Flight, and Created. One row is visible for 'MyQueue', showing 364 messages available, 0 messages in flight, and a creation date of 2015-11-13 15:10:40 GMT-05:00.

Name	Messages Available	Messages in Flight	Created
MyQueue	364	0	2015-11-13 15:10:40 GMT-05:00

The number of instances in response increases

The screenshot shows the AWS Lambda Instances console. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. A search bar labeled 'Filter by tags and attributes or search by keyword' is present. Below the header is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, and Status Checks. Several instances are listed, including 'i-e4bc125a' (running), 'i-2ac89d9d' (running), 'First Instance' (running), 'i-5ace60e4' (running), 'i-5bce60e5' (terminated), 'i-b1bfea06' (terminated), and 'i-b0bfea07' (terminated).

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	
	i-e4bc125a	t1.micro	us-east-1a	running	Initializing	
<input checked="" type="checkbox"/>	i-2ac89d9d	t1.micro	us-east-1b	running	Initializing	
	First Instance	i-9f035628	t1.micro	us-east-1b	running	2/2 checks passed
		i-5ace60e4	t1.micro	us-east-1a	running	2/2 checks passed
		i-5bce60e5	t1.micro	us-east-1a	terminated	
		i-b1bfea06	t1.micro	us-east-1b	terminated	
		i-b0bfea07	t1.micro	us-east-1b	terminated	

The number of instances also increases in Load Balancer

The screenshot shows the AWS Load Balancer console. At the top, there is a filter bar with 'Search Load Balancers'. Below the header is a table with columns: Load Balancer Name, DNS Name, and Port Configuration. One row is visible for 'LB' with a DNS name of 'LB-1106736577.us-east-1.el...' and port configuration '80 (HTTP) forward'. Below this, a section titled 'Load balancer: LB' shows tabs for 'Description', 'Instances' (which is selected), 'Health Check', 'Monitoring', 'Security', and 'List'. A note says 'Connection Draining: Disabled (Edit)'. An 'Edit Instances' button is present. A table below lists instances with columns: Instance ID, Name, and Availability Zone. Three instances are listed: 'i-e4bc125a' (us-east-1a), 'i-2ac89d9d' (us-east-1b), and 'i-5ace60e4' (us-east-1a).

Instance ID	Name	Availability Zone
i-e4bc125a		us-east-1a
i-2ac89d9d		us-east-1b
i-5ace60e4		us-east-1a

Keeps increasing instances as it is consuming messages

Filter by tags and attributes or search by keyword							
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
		i-e4bc125a	t1.micro	us-east-1a	running	2/2 checks passed	None
		i-2ac89d9d	t1.micro	us-east-1b	running	2/2 checks passed	None
	First Instance	i-9f035628	t1.micro	us-east-1b	running	2/2 checks passed	None
		i-5ace60e4	t1.micro	us-east-1a	running	2/2 checks passed	None
		i-caf0a57d	t1.micro	us-east-1b	running	Initializing	None
		i-d0f4a167	t1.micro	us-east-1b	running	2/2 checks passed	None
		i-78b21cc6	t1.micro	us-east-1a	running	Initializing	None
		i-5bce60e5	t1.micro	us-east-1a	terminated		None
		i-b1fea06	t1.micro	us-east-1b	terminated		None
		i-b0bfea07	t1.micro	us-east-1b	terminated		None

And in the Load Balancer

The screenshot shows the AWS Load Balancer console. At the top, there is a search bar and a navigation menu. Below that, there are two main fields: "Load Balancer Name" and "DNS Name". The "Load Balancer Name" field contains "LB" and the "DNS Name" field contains "LB-1106736577.us-east-1.el...".

Edit Instances

Instance ID	Name	Availability Zone
i-e4bc125a		us-east-1a
i-2ac89d9d		us-east-1b
i-62b11fdc		us-east-1a
i-5ace60e4		us-east-1a
i-caf0a57d		us-east-1b
i-d0f4a167		us-east-1b
i-78b21cc6		us-east-1a

The Alarm status changes in CloudWatch while messages are greater than the 300 threshold in scale-up alarm

Create Alarm Modify Copy Delete			
Filter: All alarms <input type="button" value="Search Alarms"/> 1 to 2 of 2 Alarms			
State	Name	Threshold	Config Status
<input checked="" type="checkbox"/> ALARM	scale_up	ApproximateNumberOfMessagesVisible >= 300 for 1 minute	No notifications
<input type="checkbox"/> OK	scale-dn	ApproximateNumberOfMessagesVisible <= 100 for 5 minutes	No notifications

After consumption there will be a low number of messages in queue

Queues			
Create New Queue Queue Actions			
Filter by Prefix:			
Name		Messages Available	Messages in Flight
MyQueue		0	0

The CloudWatch Alarm for scale-dn is raised

Filter: All alarms <input type="button" value="Search Alarms"/> 1 to 2 of 2.			
State	Name	Threshold	Config Status
<input type="checkbox"/> ALARM	scale-dn	ApproximateNumberOfMessagesVisible <= 100 for 5 minutes	No notifications
<input type="checkbox"/> OK	scale_up	ApproximateNumberOfMessagesVisible >= 300 for 1 minute	No notifications

The number of instances decrease

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	First Instance	i-9f035628	t1.micro	us-east-1b	green running	green 2/2 checks passed
		i-1a8927a4	t1.micro	us-east-1a	green running	green 2/2 checks passed
		i-e4bc125a	t1.micro	us-east-1a	red terminated	
		i-2ac89d9d	t1.micro	us-east-1b	red terminated	
<input checked="" type="checkbox"/>	i-07faafb0	i-07faafb0	t1.micro	us-east-1b	red terminated	
		i-62b11fdc	t1.micro	us-east-1a	red terminated	
		i-4fe6b3f8	t1.micro	us-east-1b	red terminated	
		i-5ace60e4	t1.micro	us-east-1a	red terminated	
		i-5bce60e5	t1.micro	us-east-1a	red terminated	
		i-caf0a57d	t1.micro	us-east-1b	red terminated	
		i-b0bfea07	t1.micro	us-east-1b	red terminated	
		i-d0f4a167	t1.micro	us-east-1b	red terminated	
		i-78b21cc6	t1.micro	us-east-1a	red terminated	

Also decreases in Load Balancer

Create Load Balancer Actions ▾

Filter: Search Load Balancers X

<input checked="" type="checkbox"/>	Load Balancer Name	DNS Name	Port Config
<input checked="" type="checkbox"/>	LB	LB-1106736577.us-east-1.el...	80 (HTTP) f

Load balancer: LB

Description Instances Health Check Monitoring Security ⋮

Connection Draining: Disabled ([Edit](#))

[Edit Instances](#)

Instance ID	Name	Availability Zone
i-1a8927a4		us-east-1a

[Edit Availability Zones](#)

And in the Autoscaling Group

The screenshot shows the AWS Auto Scaling Groups page. At the top, there is a blue button labeled "Create Auto Scaling group" and a "Actions" dropdown menu. Below this is a search bar labeled "Filter: Filter Auto Scaling groups..." with a clear button "X". A table header row includes columns for Name, Launch Configuration, Instances, Desired, Min, Max, and Available. A single row is shown for the group "asg", which uses launch configuration "lc", has 1 instance, and is desired at 1, with a minimum of 1 and a maximum of 10. The "Available" column shows "us-east".

Auto Scaling Group: asg

Details Activity History Scaling Policies Instances Notifications Tags

Actions

Filter: Any Health Status ▾ Any Lifecycle State ▾ Filter instances...

	Instance ID	Lifecycle	Launch Configuration Name
<input type="checkbox"/>	i-1a8927a4	InService	lc

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.

Total Points: 30

Add Policy to new Autoscaling group asg_2 to increase to 3 instances at 4:00pm (21:00 UTC)

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling put-scheduled-update-group-action --scheduled-action-name SchedScaleUp --auto-scaling-group-name asg_2 --start-time "2015-11-12T21:00:00Z" --desired-capacity 3
Marnie's-MacBook-Air:~ marnie$
```

Add

Policy to new Autoscaling group asg_2 to decrease to 1 instance at 5:00pm (22:00 UTC)

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling put-scheduled-update-group-action --scheduled-action-name SchedScaleDown --auto-scaling-group-name asg_2 --start-time "2015-11-12T22:00:00Z" --desired-capacity 1
Marnie's-MacBook-Air:~ marnie$
```

Instances after 4:00pm (scheduled time to have 3)

Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status
i-674a07d0	InService	lc	us-east-1b	Healthy
i-4c7d30fb	InService	lc	us-east-1b	Healthy
i-4d7d30fa	InService	lc	us-east-1b	Healthy

Instances after 5:00pm (00:00:00 UTC)

The screenshot shows the AWS Auto Scaling console. At the top, there's a 'Create Auto Scaling group' button and an 'Actions' dropdown. Below that is a search bar labeled 'Filter: Filter Auto Scaling groups...'. A table lists two Auto Scaling Groups: 'asg_2' and 'asg'. The 'asg_2' row has a blue selection indicator. The table columns include Name, Launch Configuration, Instances, Desired, Min, Max, Availability Zones, and Default Cooldown.

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown
asg_2	lc	1	1	1	3	us-east-1b	300
asg	lc	1	1	1	10	us-east-1b, us-east-1a	300

Below the groups, a section titled 'Auto Scaling Group: asg_2' shows one instance. It includes tabs for Details, Activity History, Scaling Policies, Instances (which is selected), Notifications, and Tags. An 'Actions' dropdown is also present here.

Further down, another table shows the details of the single instance in the 'asg_2' group. It includes filters for Health Status and Lifecycle State, and a search bar for 'Filter instances...'. The instance details table has columns for Instance ID, Lifecycle, Launch Configuration Name, Availability Zone, and Health Status.

Instance ID	Lifecycle	Launch Configuration Name	Availability Zone	Health Status
i-4d7d30fa	InService	lc	us-east-1b	Healthy

Problem 3:

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

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.

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

Try deleting Instances

One instance should be able to delete since it is not in an autoscaling group. The other two instances each belong to AutoScaling groups with a minimum of 1.

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, and Elastic Block Store. The main area displays a table of instances. One instance, 'First Instance' (ID: i-3aa5c28d), is highlighted. A context menu is open over this instance, with 'Terminate' being the selected option under the 'Actions' dropdown. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, and Public IP. The 'Instance State' column shows various statuses: running, terminated, and so on. The 'Public DNS' and 'Public IP' columns show the specific addresses for each instance.

They all terminated

This screenshot shows the same EC2 Management Console interface after the instances have been terminated. The table now lists five instances, all of which are in the 'terminated' state. The 'Instance State' column for each row is colored red, indicating they are no longer active. The rest of the table structure remains the same, with columns for Name, Instance ID, Instance Type, Availability Zone, Status Checks, Alarm Status, Public DNS, and Public IP.

But then instances start being created

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with various navigation links like EC2 Dashboard, Instances, AMIs, and Network & Security. The main content area is titled "Create Auto Scaling group" and shows a table of Auto Scaling Groups. One group, "asg_2", is listed with a launch configuration "lc", 1 instance, and 1 desired instance. Another group, "asg", is also listed with a launch configuration "lc", 1 instance, and 1 desired instance. Below this, under "Auto Scaling Group: asg", there's a detailed view of the single instance. The instance ID is "i-3502a48b", it's in a "Pending" lifecycle state, and it's associated with the "lc" launch configuration, located in the "us-east-1a" availability zone and is "Healthy".

This screenshot is from the same session as the previous one, but it shows a different state. The "asg_2" group now has one instance in a "Pending" state, while the "asg" group remains at zero instances. The rest of the interface and data are identical to the first screenshot.

Modify both autoscaling groups to have minimum and maximum size set to zero

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling update-auto-scaling-group --auto-scaling-group-name asg --min-size 0 --max-size 0
Marnie's-MacBook-Air:~ marnie$ aws autoscaling update-auto-scaling-group --auto-scaling-group-name asg_2 --min-size 0 --max-size 0
Marnie's-MacBook-Air:~ marnie$
```

Delete Autoscaling Groups

```
Marnie's-MacBook-Air:~ marnie$ aws autoscaling delete-auto-scaling-group --auto-scaling-group-name asg
Marnie's-MacBook-Air:~ marnie$ aws autoscaling delete-auto-scaling-group --auto-scaling-group-name asg_2
Marnie's-MacBook-Air:~ marnie$
```

Delete all instances again

The screenshot shows the AWS EC2 Management Console interface. On the left, there is a navigation sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, AMIs, and Bundle Tasks. The Instances section is currently selected. The main content area displays a table of terminated instances. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS, and Public IP. There are seven instances listed, all of which are terminated. A message at the bottom of the table says "Select an instance above". At the bottom of the page, there are links for Feedback, English, Privacy Policy, and Terms of Use.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
i-4d7d30fa	t1.micro	us-east-1b	terminated	None	None	None	None	None
i-4cd300fb	t1.micro	us-east-1b	terminated	None	None	None	None	None
First Instance	i-3aa5c28d	t1.micro	us-east-1b	terminated	None	None	None	None
i-ded07680	t1.micro	us-east-1a	terminated	None	None	None	None	None
i-3502a48b	t1.micro	us-east-1a	terminated	None	None	None	None	None
i-b1125e06	t1.micro	us-east-1b	terminated	None	None	None	None	None
scheduledSc...	i-674a07d0	t1.micro	us-east-1b	terminated	None	None	None	None

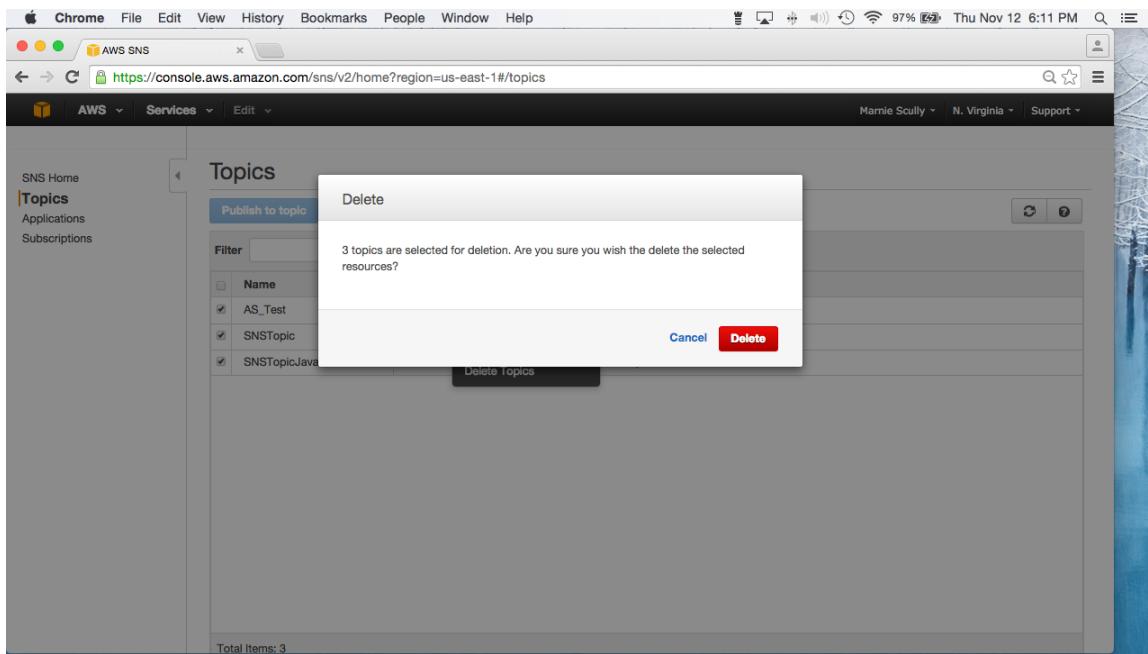
Delete AMI

The screenshot shows the AWS EC2 Management Console. On the left, the navigation pane includes options like Instances, Images (selected), and Elastic Block Store. The main content area displays a table of AMIs, with one row selected: "My AMI" (ami-62611908). A modal dialog titled "Deregister" is open, asking "Are you sure you want to deregister these images? ami-62611908 - My AMI". Below the table, there's a detailed view of the selected AMI, showing its ID, owner, name, and source.

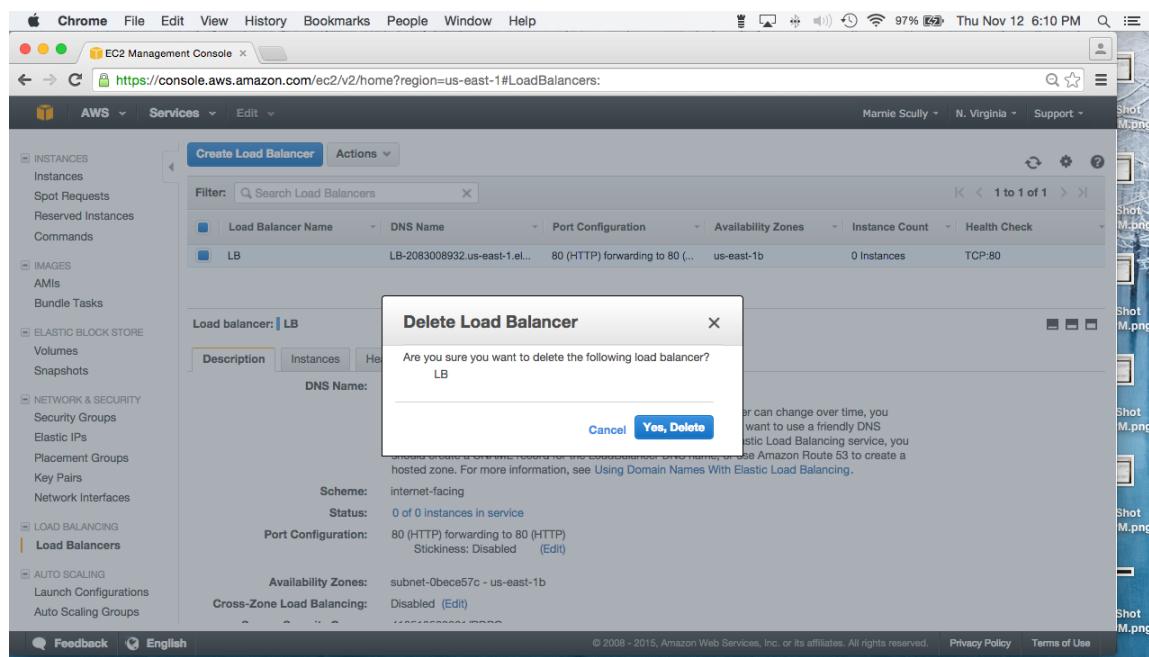
Delete Queue

The screenshot shows the AWS SQS Management Console. The left sidebar lists "Queues" and "Create New Queue". The main area shows a table with one item: "MyQueue". A modal dialog titled "Delete Queues" is open, asking "Are you sure you want to delete the following queue, and any messages left in it? • MyQueue - contains no messages.". At the bottom of the page, there's a detailed view of the queue, including its URL, ARN, and various metrics like message counts and visibility timeout.

Delete SNS topic



Delete Load Balancer



Delete Launch Configuration

The screenshot shows the AWS EC2 Management Console. In the top navigation bar, the 'Actions' dropdown is open, with 'Delete launch configuration' highlighted. Below this, a table lists a single launch configuration named 'lc'. The table includes columns for Name, AMI ID, Instance Type, Spot Price, and Creation Time. The 'lc' entry has values: ami-62611908, t1.micro, and November 12, 2015 1:40:18 PM... The 'Delete launch configuration' option is also visible in the Actions dropdown for this specific row.

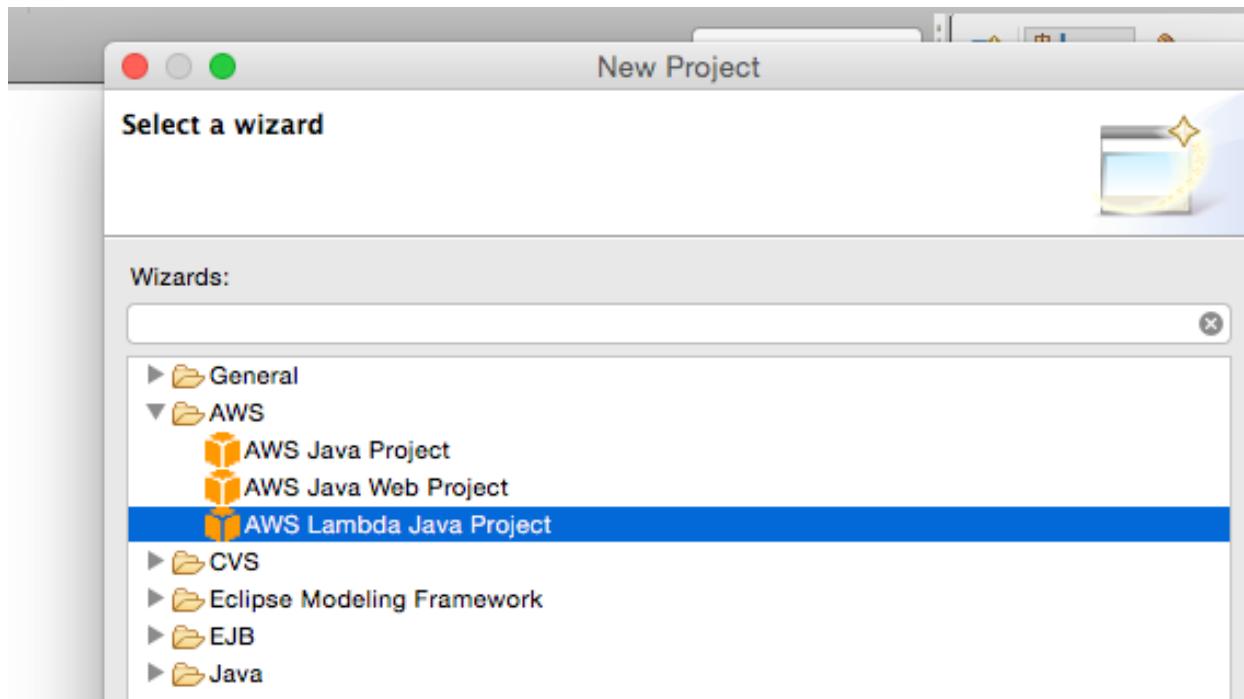
Delete Alarms

The screenshot shows the AWS CloudWatch Management Console. The left sidebar is expanded to show the 'Alarms' section, which is currently selected. A single alarm named 'scale-dn' is listed under the 'ALARM' category. The main pane displays the details of this alarm, including its state (ALARM), name (scale-dn), threshold (ApproximateNumberOfMessagesVisible <= 100 for 1 minute), and config status (Invalid notification). A modal dialog box titled 'Delete Alarms' is open in the center, asking 'Are you sure you want to delete this alarm?'. The 'scale-dn' alarm is listed in the dialog. At the bottom of the dialog are 'Cancel' and 'Yes, Delete' buttons. The background shows the detailed configuration for the 'scale-dn' alarm, including its state history, description (Scale down Instances when less messages in queue), threshold (ApproximateNumberOfMessagesVisible <= 100 for 1 minute), and actions (In ALARM: Your notification topic: AS_Test could not be found. It may have been deleted. Please use a).

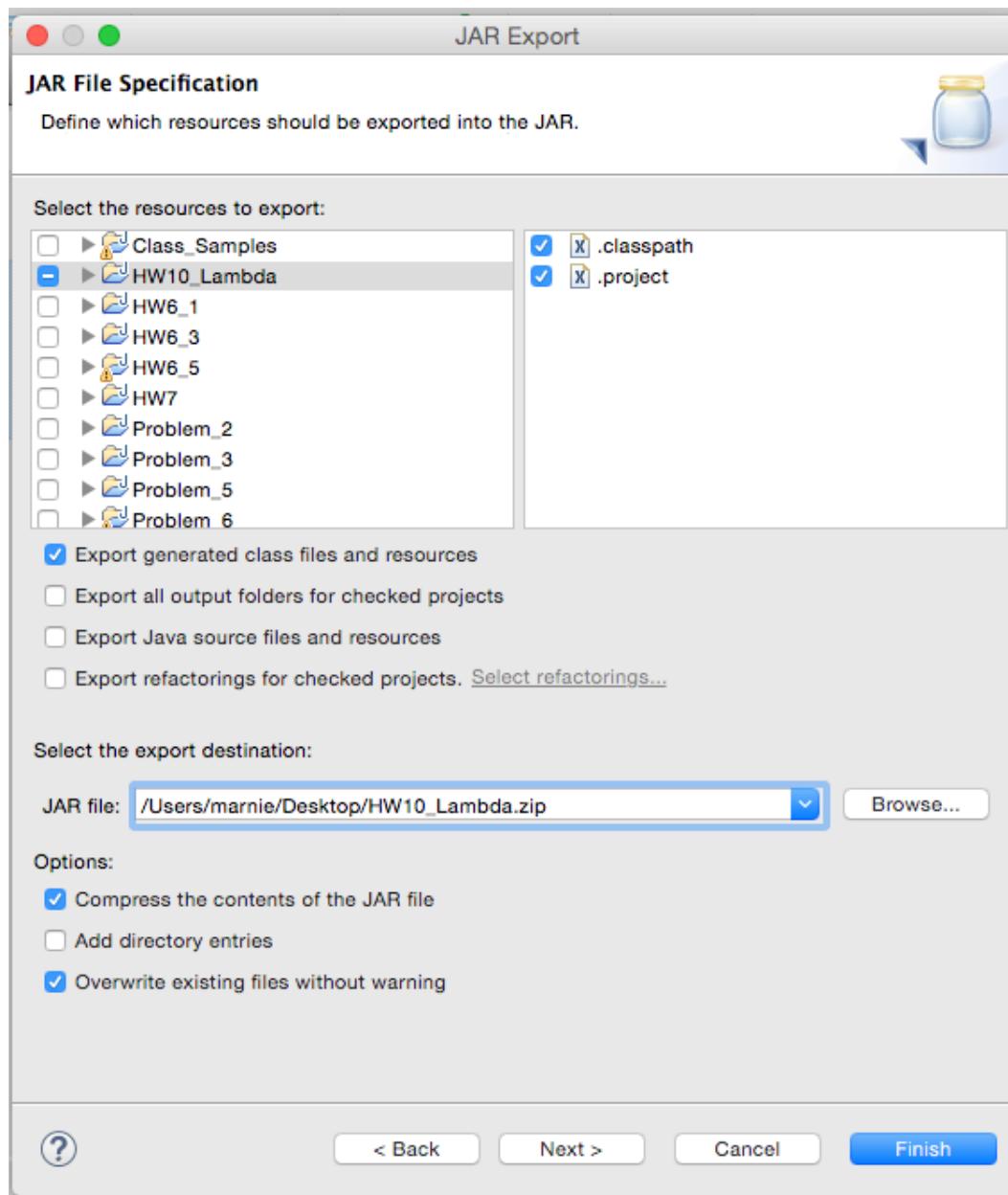
Problem 4.

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. string string to the output. Demonstrate that your lambda function works.

Total Points: 25

Create an AWS Lambda Java Project in Eclipse**Create a function that returns a string that was input**

```
public class EchoString {  
    public String echoHandler(String str, Context context){  
        LambdaLogger logger = context.getLogger();  
        logger.log("received : " + str);  
        return str;  
    }  
}
```

Export project from Eclipse into zip file

Create Administrator group and add user admin to it

Set Group Name

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

Group Name: Administrators

Example: Developers or ProjectAlpha

Maximum 128 characters

Attach Policy

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

Filter: Policy Type ▾					Showing 172 results
	Policy Name	Attached Entities	Creation Time	Edited Time	
<input checked="" type="checkbox"/>	 AdministratorAccess	0	2015-02-06 13:39 EST	2015-02-06 13:39 EST	
<input type="checkbox"/>	 AmazonAPIGatewayAdmin...	0	2015-07-09 13:34 EDT	2015-07-09 13:34 EDT	
<input type="checkbox"/>	 AmazonAPIGatewayInvoke	0	2015-07-09 13:36 EDT	2015-07-09 13:36 EDT	

IAM > Groups > Administrators

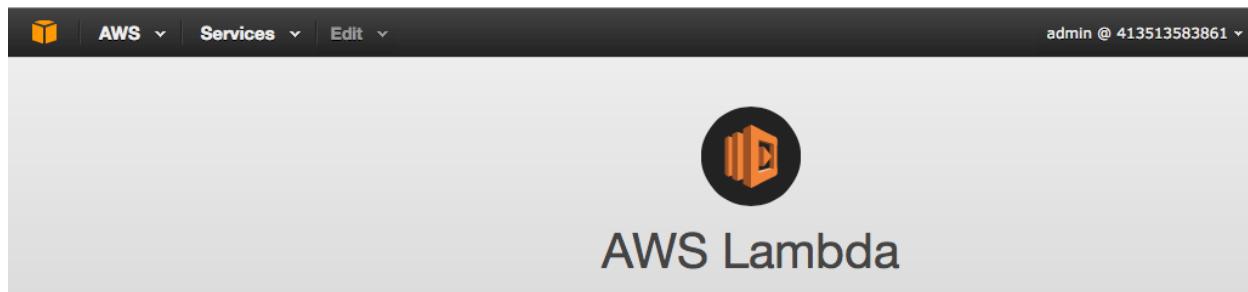
◀ ▾ Summary

Group ARN: arn:aws:iam::413513583861:group/Administrators
Users (in this group): 1
Path: /
Creation Time: 2015-11-12 16:50 EST

Users **Permissions**

This view shows all users in this group: **1 User**

User	Actions
 admin	Remove User from Group

Sign in as this user**Create a Lambda Function and skip the blueprint****Upload the zip file****Configure function**

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

Name*	<input type="text" value="HW10Lambda"/>
Description	<input type="text" value="Lambda Function that Echos string"/>
Runtime*	<input type="text" value="Java 8"/> ▼

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.

Upload HW10_Lambda.zip

Lambda function handler and role

Handler*	<input type="text" value="EchoString::echoHandler"/>
Role*	<input type="text" value="lambda_basic_execution"/> ▼ ⓘ

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

Advanced settings

The screenshot shows the AWS Lambda console interface. At the top, it displays the path "Lambda > Functions > HW10Lambda" and the ARN "arn:aws:lambda:us-east-1:413513583861:function:HW10Lambda". Below this, there are three buttons: "Save", "Save and test" (which is highlighted in blue), and "Actions". A success message box states "Congratulations! Your Lambda function "HW10Lambda" has been successfully created." Below the message are tabs for "Code", "Configuration", "Event sources", "API endpoints", and "Monitoring". A note indicates that inline editing is not possible due to file size or content. It provides options to upload a local ZIP file or an S3 object. A file named "HW10_Lambda.zip" is shown being uploaded.

Test the Lambda Function

The screenshot shows the "Input test event" section of the Lambda function configuration. It includes a "Sample event template" dropdown and a code editor area. The code editor contains the following JSON template:

```
1  "echoMe"
```

The Result

Execution result: succeeded (logs)

"echoMe"

Summary	
Request ID	11cb5b3d-898a-11e5-8643-0724db001dca
Duration	87.66 ms
Billed duration	100 ms
Resources configured	512 MB
Max memory used	66 MB

Log output

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.

```
START RequestId: 11cb5b3d-898a-11e5-8643-0724db001dca Version: $LATEST
received : echoMeEND RequestId: 11cb5b3d-898a-11e5-8643-0724db001dca
REPORT RequestId: 11cb5b3d-898a-11e5-8643-0724db001dca Duration: 87.66 ms      Billed Duration: 100 ms
Memory Size: 512 MB     Max Memory Used: 66 MB
```
