Systems Operations on AWS
Lab Guide

Version 1.1 - Update 1

100-SYS-11-EN-LG

Copyright © 2013, 2014 Amazon Web Services, Inc. and its affiliates.
All rights reserved.

This work may not be reproduced or redistributed, in whole or in part,
without prior written permission from Amazon Web Services, Inc.

Commercial copying, lending, or selling is prohibited.

Corrections or feedback on the course, please email us at:
aws-course-feedback@amazon.com.

For all other questions, please email us at aws-training-info@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Table of Contents

Login to the AWS Management Console **4**

Lab: Virtual Private Cloud & CloudFormation 6

Linux Labs **19**
- Elastic Compute Cloud (EC2) 20
- Elastic Block Store (EBS) 33
- Tagging 51
- Monitoring 59
- Backup and Archive 71
- Security – Event Collection 88
- Logging 100
- Creating an Elastic Infrastructure 113

Windows Labs **134**
- Elastic Compute Cloud (EC2) 135
- Elastic Block Store (EBS) 147
- Tagging 165
- Monitoring 176
- Security – Event Collection 192
- Logging 205
- Creating an Elastic Infrastructure 224

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Login to the AWS Management Console

Using qwikLAB to login to the AWS Management Console

The first step is for you to login to Amazon Web Services.

:

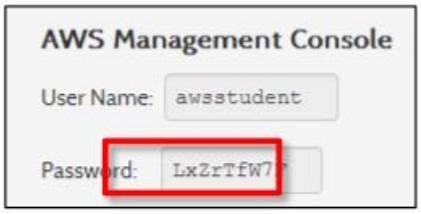
1. To the right of the lab title, click the **Start Lab** button to launch your *qwikLAB*™. If you are prompted for a token, use the one distributed to you (or the token you purchased).

Note: A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.

2. On the lab details page, notice the lab properties.
   a. **Duration -** The time the lab will run before automatically shutting down.
   b. **Setup Time -** The estimated time to set up the lab environment.
   c. **AWS Region** - The AWS Region in which the lab resources are created.

Note: The AWS Region for your lab will differ depending on your location and the lab setup.

3. In the AWS Management Console section of the *qwikLAB*™ page, copy the **Password** to the clipboard.



4. Click the **Open Console** button.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

5. Log into the AWS Management Console using the following steps.
   a. In the **User Name** field type **awsstudent**.
   b. In the **Password** field, paste the password copied from the lab details page.
   c. Click **Sign in using our secure server**.

Note: The AWS account is automatically generated by *qwikLAB*™. Also, the login credentials for the awsstudent account are provisioned by *qwikLAB*™ using AWS Identity Access Management.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
VPC & CloudFormation
Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

In this lab, you will use two different techniques to provision VPC scaffolding. First, you are going to do all the necessary steps to manually create the VPC through the AWS Management Console. After that, you are going to use CloudFormation to automate the whole process.
During this lab, you will:

- Manually create a VPC
- Configure subnets in the VPC
- Create and attach an Internet Gateway (IGW)
- Define routing tables and routes
- Setup security groups and rules to control access
- Deploy a NAT instance
- Deploy a VPC using a CloudFormation template

The VPC we are going to create is a small and simple one, but will contain many different important elements of real-world, large-scale VPCs. It will contain **2 subnets**: a *public subnet* and a *private subnet*.

Instances launched into a **public subnet** will be able to receive traffic from the Internet and directly connect to other hosts on the Internet. For this, they will be assigned each an **Elastic IP** addresses, which is publicly routable, and the subnet will have a **Routing Table** with a **Route** pointing to an **Internet Gateway** that will be launched into the VPC.
On the other hand, instances on a **private subnet** won't have public IP addresses. For them to be able to connect to other hosts on the Internet (including AWS service endpoints), you'll deploy a **NAT instance** on the public subnet and set in the private subnet a Routing Table with a Route to the NAT Instance.

Finally, the traffic between these two subnets is routed through a **local router**, which is represented by a route to *local* on a routing table.

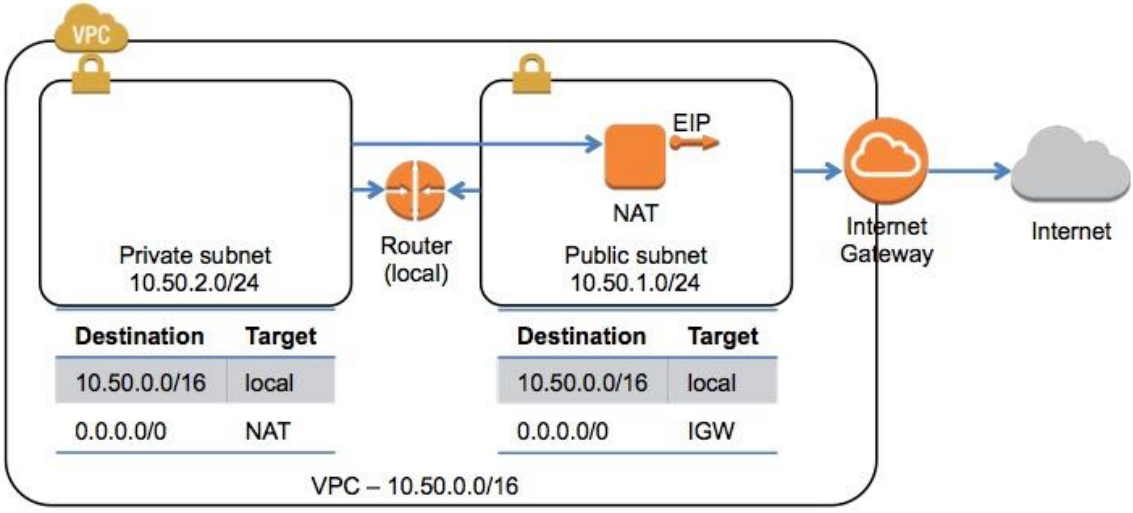


Figure 1 - Network architecture (goal of this lab)

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Technical Knowledge Prerequisites
To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line tools.

Topics Covered
The lecture component of this module took us through the various aspects of a VPC. This lab exercise will reinforce those concepts, specifically:

- VPC and subnet definitions
- Internet Gateway (IGW) usage
- Routing traffic within the VPC
- Security groups and rules to control access

Technical Knowledge Prerequisites
To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line tools.

Topics Covered
The lecture component of this module took us through the various aspects of a VPC. This lab exercise will reinforce those concepts, specifically:

- VPC and subnet definitions
- Internet Gateway (IGW) usage
- Routing traffic within the VPC
- Security groups and rules to control access

Provision the Virtual Network Scaffolding (Manually)
Use the AWS Management Console to create a VPC and two subnets.

Identify the Names of the Availability Zones
A subnet exists inside a single availability zone. In some cases AWS can automatically choose an availability zone for you. Often, however, you will want to choose your own availability zones, to ensure your VPC spans multiple ones and your service is highly available.
The name of the availability zones depend on your account. They are not fixed – they change from account to account. The following steps will help you identify the names of your availability zones.

1. In the AWS Management Console choose **Services > All AWS Services > EC2**.
2. On the EC2 dashboard scroll down to **Service Health >Availability Zone Status**
3. Note the names of the availability zones from the list. You will be referring to this list later in this lab.

Create the VPC

4. In the AWS Management Console choose **Services > All AWS Services > VPC**.
5. Click the **`Your VPCs`** link from the left menu.
6. Click the **`Create VPC`** button.
7. In the **`Create VPC`** dialog box, enter a **`CIDR Block`** of **`10.50.0.0/16`**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

8. Leave the Tenancy **`Default`**
9. Click **`Yes, Create`**.

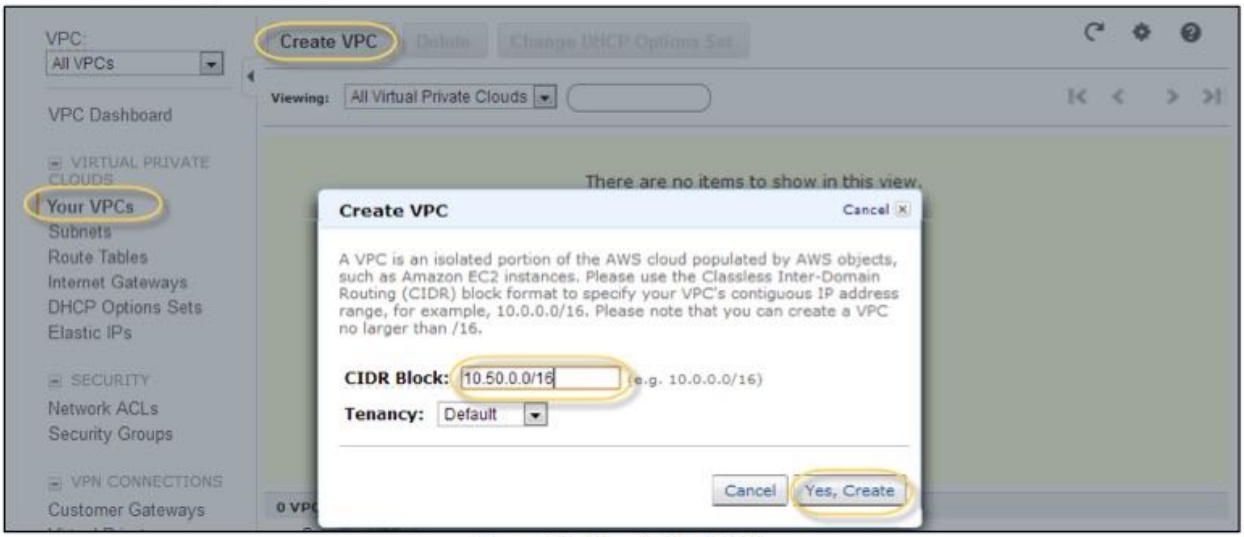


Figure 2 - Create the VPC

10. Make a note of the VPC ID.

Create the subnets

In the next sets of steps you are going to create 2 subnets in your VPC. They will be called *public* and *private*, but what will make they behave differently will be configured later on – when you add Route Tables.

The process is the same for bigger VPCs. This is what it looks like when creating a subnet:

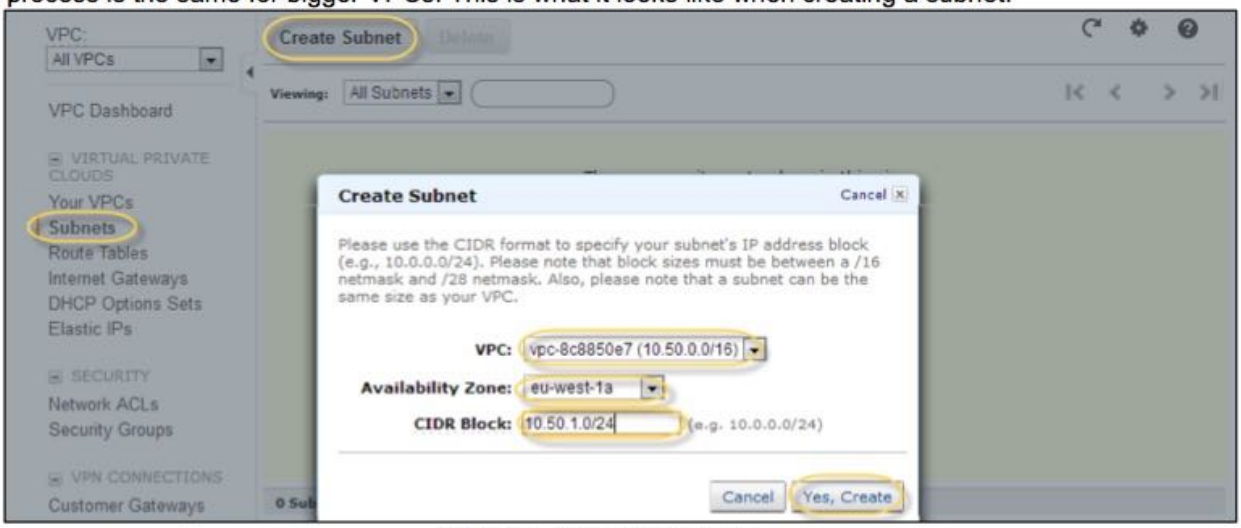


Figure 3 – Creating subnets

Create the Public Subnet

11. Select the **`Subnets`** link from the left hand menu.
12. Click the **`Create Subnet`** button.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

13. Ensure that the VPC is set to the one created earlier, by confirming either the VPC ID or the CIDR block.
14. Choose an Availability Zone. Make note of this Availability Zone. You will reference it later in the lab.
15. Specify a CIDR Block of **10.50.1.0/24**.
16. Click **`Yes, Create`**.

Create the Private Subnet

17. Click the **`Create Subnet`** button.
18. Ensure that the VPC is set to the one created earlier, by confirming either the VPC ID or the CIDR block.
19. Choose the *same* availability zone as the one chosen for the **Public Subnet**.
20. Specify a CIDR Block of **10.50.2.0/24**.
21. Click **`Yes, Create`**.

Internet Gateway (IGW)

The Internet Gateway (IGW) is the component of a VPC that handles Internet-facing traffic. You generally only need one IGW for your whole VPC if you intend to launch instances with publicly routable IP addresses that will receive traffic from or send traffic to the Internet. Note that this includes accessing many of the regional services offered by AWS, such as DynamoDB, SQS, SNS, CloudSearch, etc.

Create the Internet Gateway

22. Select the **`Internet Gateways`** link from the left menu.
23. Click the **`Create Internet Gateway`** button.
24. Click the **`Yes, Create`** button to create the IGW.

Attach the Internet Gateway

25. Select the Internet Gateway you just created and click the **`Attach to VPC`** button.
26. Ensure that the VPC is set to the one created earlier, by confirming either the VPC ID or the CIDR block.
27. Click the **`Yes, Attach`** button to attach the IGW.

Route Tables

Every VPC comes with a *Main* route table, which is attached to every newly created subnet by default. It is identified by the value *Yes* on the *Main* column of the list of route tables. It contains a default route (i.e., a route with destination 0.0.0.0/0), which targets the *local router* – an abstract component of a VPC that routes inter-subnet traffic. We usually create more specific route table, and leave the Main one alone.

In the next steps, you'll create the *public* and the *private* route tables. You'll see that the difference between them is their *default route*: in the public subnet, it targets the Internet Gateway; in the private one, it targets a NAT instance that you'll later launch into the public subnet.

After the steps, you will find a few figures illustrating the process.

Create the Public Route Table

28. Select the **`Route Tables`** link from the left menu.
29. Click the **`Create Route Table`** button.
30. Ensure that the VPC is set to the one created earlier, by confirming either the VPC ID or the CIDR block.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

31. Click the **`Yes, Create`** button to create the route table.

Add a default route to the Public Route Table targeting the IGW

32. Select the route table just created and click the **`Routes`** tab below (note: it shows "no" on column *Main*) and be sure it is the only one selected.
33. Add a route with a **Destination** of **`0.0.0.0/0`** and with **Target** set to the Internet Gateway.
34. Click the **`Add`** button.
35. Click the **`Yes, Create`** button to create the route.

Associate the Public Subnet with the Public Route Table

36. Click the **`Associations`** tab next to the **`Routes`** tab.
37. Select the item that corresponds with the Public Subnet (**`10.50.1.0/24`**).
38. Click the **`Associate`** button.
39. Click the **`Yes, Associate`** button to associate the subnet with the public route table.

Tag the Public Route Table

40. Click the **`Tags`** tab next to the **`Route Propagation`** and **`Associations`** tabs.
41. Click the **`Add/Edit Tags`** button.
42. In the **`Value`** column of the **`Name`** key (tag); type **Public**.
43. Click the **`Save Tags`** button.

Create the Private Route Table

44. Click the **`Create Route Table`** button.
45. Ensure that the VPC is set to the one created earlier, by confirming either the VPC ID or the CIDR block.
46. Click the **`Yes, Create`** button to create the route table.

Add a default route to the Private Route Table targeting a NAT instance

Whoa – not just yet! Although we did create a route for the public route table we already had a target; i.e. the Internet Gateway we created earlier! We will be creating a route for the private route table; but only *after* we create a NAT instance (which we will get to later in the lab). For now we will proceed straight to the subnet association for the private route table.

Associate the Private Subnet with the Private Route Table

47. Select the route table you just created (the table associated with 0 subnets which shows "no" on the *Main* column) and be sure it is the only one selected.
48. Click the **`Associations`** tab next to the **`Routes`** tab.
49. Select the item that corresponds with the Private Subnet (**`10.50.2.0/24`**).
50. Click the **`Associate`** button.
51. Click the **`Yes, Associate`** button to associate the subnet with the route table.

Tag the Private Route Table

52. Click the **`Tags`** tab next to the **`Route Propagation`** and **`Associations`** tabs.
53. Click the **`Add/Edit Tags`** button.
54. In the **`Value`** column of the **`Name`** key (tag); type **Private**.
55. Click the **`Save Tags`** button.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Figure 4 - Create a route table

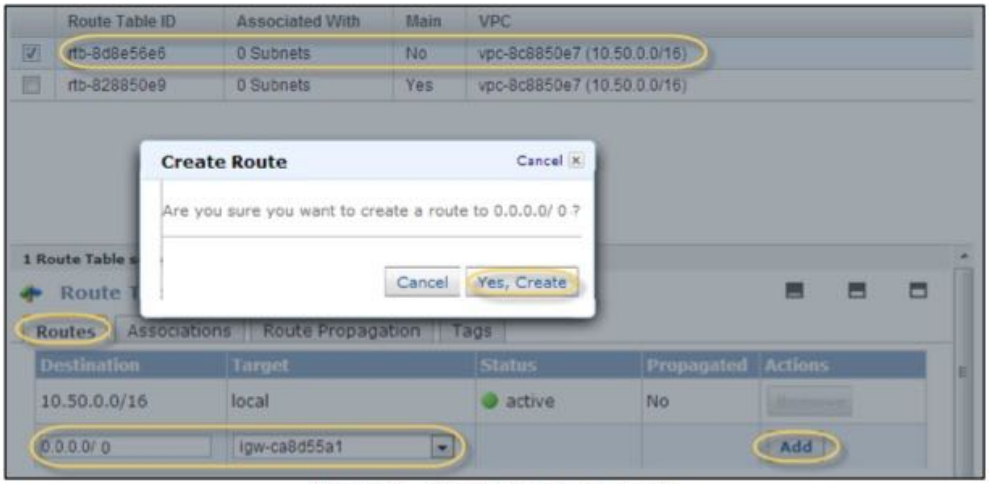


Figure 5 - Create a default route

Figure 6 - Associate a route table with a subnet

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Elastic IP Address (EIP)

An Elastic IP Address (EIP) is a publicly routable IP address that you can freely attach and reattach to your instances. It is a requirement for an instance to be accessible from the Internet (in addition to having an Internet Gateway and properly configured routes, which we did in the previous parts of this lab).

In this lab, the EIP you are going to allocate will be associated with the NAT instance to allow it to access the Internet.

Note: when allocating an EIP, you *might* have to specify where it will be used: either in VPC or in EC2 (classic). Some regions or newer AWS accounts always use VPC (and don't have EC2-Classic anymore), and, in this case, you won't be asked where it will be used.

Allocate an Elastic IP Address

56. Click the **`Elastic IPs`** link from the left menu.
57. Click the **`Allocate New Address`** button.
58. If there's a dropdown on the confirmation window, ensure that the EIP will be used in **VPC**.
59. Click the **`Yes, Allocate`** button to create the EIP.

Security Groups

We will now create the various Security Groups that will be used to secure our VPC. Later we will associate these Security Groups with various instances such as our NAT instance.

Create the Elastic Load Balancer (ELB) Security Group

60. Click the **`Security Groups`** link from the left menu.
61. Click the **`Create Security Group`** button.
62. Name the Security Group **`ELBSecurityGroup`** and provide an appropriate description.
63. Ensure that the VPC is set to the one created earlier, by confirming the VPC ID.
64. Click the **`Yes, Create`** button to create the Security Group.

Create the Elastic Load Balancer Inbound Rule

65. Select the ELB Security Group you just created, **`ELBSecurityGroup`**.
66. Click the **`Inbound`** tab.
67. From the **`Create a new rule`** dropdown, select **`HTTP`**.
68. Specify the source as **`0.0.0.0/0`** (everywhere, including the Internet).
69. Click the **`Add Rule`** button.
70. Click the **`Apply Rule Changes`** button.

Create the NAT Security Group

71. Click the **`Create Security Group`** button.
72. Name the Security Group **`NATSecurityGroup`** and provide an appropriate description.
73. Ensure that the VPC is set to the one created earlier, by confirming the VPC ID.
74. Click the **`Yes, Create`** button to create the Security Group.

Create the NAT Inbound Rules

75. Select the NAT Security Group just created, **`NATSecurityGroup`**.
76. Click the **`Inbound`** tab.
77. From the **`Create a new rule`** dropdown, select **`All Traffic`**.
78. Specify the source as **`10.50.2.0/24`**, the Private Subnet

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

79. Click the **Add Rule** button.
80. Click the **Apply Rule Changes** button.

Create Web Tier Security Group

81. Click the **Create Security Group** button.
82. Name the Security Group **WebTierSecurityGroup** and provide an appropriate description.
83. Ensure that the VPC is set to the one created earlier, by confirming the VPC ID.
84. Click the **Yes, Create** button to create the Security Group.

Create the Web Tier Inbound Rules

85. Select the **ELB Security Group** and on the **Details** tab, note the **Group ID**.
86. Select the **Web Tier Security Group** just created.
87. Click the **Inbound** tab.
88. From the **Create a new rule** dropdown, select **HTTP**.
89. Specify the source as the ID of the ELB Security Group you obtained in step 1
    a. Note: the format is `sg-xxxxxx`.
    b. Note: once you start typing `sg-`, autocompletion will suggest the available SG IDs.
90. Click the **Apply Rule Changes** button.

Provision the NAT Instance in Amazon EC2

The NAT instance will provide Internet connectivity to instances in the private subnet. Since it will need to be able to connect to the internet, it needs to be placed on the public subnet and will need a publicly routable IP address – the EIP we allocated a few steps ago.
You'll have to ensure that the Kernel is configured to forward IPv4 traffic and that IPTables is configured to do NAT.

Also, you'll have to disable a security feature of EC2 that comes enabled by default: Source / Dest check. This is kind of additional firewall around the instance that would drop IPv4 packets which are not destined to the instance's IP. Since this instance will act as a NAT, its main job is to forward IP packets that are destined to other instances, which explains why you need to disable the Source / Dest check.

Deploy the NAT Instance

Launch the NAT Instance

91. Choose **Services > All AWS Services > EC2**.
92. Click the **Instances** link from the left menu.
93. Click the **Launch Instance** button.
94. Ensure the **Quick Start** tab is selected on the left side.
95. Locate **Amazon Linux AMI 20xx.yy.z** in the list of images (it should be the first one).
96. Click on the **Select** button for that AMI.
97. Click the **General purpose** tab on the left menu.
98. Click **m1.small** from the list of instance types.
99. Click the **Next: Configure Instance Details** button.
100. On the **Network** dropdown, select the VPC that you created earlier (note the CIDR 10.50.0.0/16)
101. On the **Subnet** dropdown (that appears right after finishing the previous step), select the **public subnet** you created earlier (note the CIDS **10.50.1.0/24**).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

102. Leave the **Public IP** checkbox unchecked – in a few steps, once the instance is launched, you will assign this instance the EIP you allocated earlier.
103. Click the **Advanced Details** section to open it. Note: you might need to scroll the page all the way down to find it.
104. On the **User data** option, ensure the radio button **As text** is selected. Note: you might need to scroll down even more to find it.
105. Still on the **User data** option, locate the text area and paste into it the contents of the following link:
https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc/static/nat-configure.sh

a. Note: **copy the script directly from the browser into the text area** – don't user MS Word or equivalent as a temporary place to paste it, otherwise the script will probably get broken and you will have to restart all this section!

106. Click the **5. Tag instance** link on the top of the page.
107. In the **Value** column of the **Name** key (tag); type **NAT**.
108. Click the **6. Configure Security Group** link on the top of the page.
109. Click the **Select an existing security group** radio button.
110. Click the **NATSecurityGroup** from the list that appears after completing the previous step. This is the security group you previously created and configured.
111. Click the **Review and Launch** button.
112. Click the **Launch** button.
113. Select **Choose an existing key pair** on the first dropdown
114. Select and select the key pair created by qwikLAB on the second dropdown. Its name should be similar to **qwiklab-1123-45678**.
115. Check the checkbox **I acknowledge that [...]**.
116. Click the **Launch Instances** button.
117. Scroll down and click the **View Instances** button.
118. Right-click the **NAT** instance and choose **Change Source/Dest. Check** from the context menu.

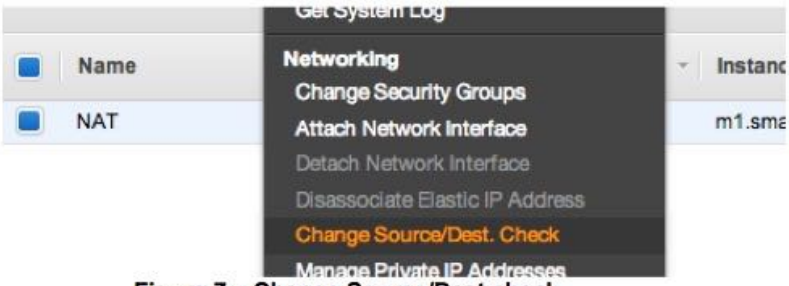


Figure 7 – Change Source/Dest check

119. Click the **Yes, Disable** button to turn off Source/Dest checking.

Associate the EIP to the Instance

120. Click the **Elastic IPs** link on the left side menu. You might need to scroll down the left menu to find it.
121. Select the address previously created. Ensure it is the correct one by noticing the column Instance ID is empty, and that the Scope is vpc.
122. Click the **Associate Address** button.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

123. Select the **NAT** instance from the drop-down list.
124. Click the **`Associate`** button to associate the EIP with the instance.

Configure the Private Route
Previously you created a route table and associated it with all of the private subnets. At the time we could not create the route since you did not have a target for the rule. Now that the NAT instance is deployed, it is the target of all Internet-bound traffic originating from the private subnet. To configure the route:

125. Choose **Services > All AWS Services > VPC** (or click VPC in the History section).
126. Click the **`Route Tables`** link on the left side menu.
127. Select the **`Private`** route table.
128. On the **Routes** tab, add a route with a Destination of **`0.0.0.0/0`** and for the Target, choose **Enter instance ID**, select the **NAT** instance and click **OK**.
129. Click the **`Add`** button to add the route to the private route table, then click **`Yes, Create`**.

Provision the Virtual Network Scaffolding (CloudFormation)

You have just created a VPC, two subnets, routing and security constructs, and an Amazon EC2 instance for NAT purposes. You may have noticed there were several steps required to do this and several places where you could have made a mistake (maybe you did!).

If you wanted to do the same again in another region, or if you just want to duplicate your virtual networking environment, the steps would have to be manually followed each time, increasing the risk of mistakes.

Introducing CloudFormation! CloudFormation uses a template (in JSON format) that describes various AWS resources and provisions them as a 'stack'. There are several advantages to this:

- Stacks can be rolled-back if an error occurs
- Deleting a stack ensures all resources are appropriately cleaned up
- Stacks can be built upon by 'updating' them with additional CloudFormation templates
- The templates support parameterization and parallel processing
- Templates lend themselves well to version control and re-usability
- Stack provisioning is consistent every time

For this part of the lab, you will provision the same environment you created manually (albeit with different resource IDs), but you will use CloudFormation and a CloudFormation template. The completed template is located here: https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc/sys-ops-lab-vpc.template

130. Gather the information needed to deploy your CloudFormation template:
    a. In the **VPC Dashboard**, click **Subnets**.
    b. Record the Availability Zone in which you have placed your subnets.
    c. Choose **Services > All AWS Services > EC2**.
    d. Click **Key Pairs**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

   e. Record the name of the qwiklab generated key pair.

131. Choose **Services > All AWS Services > CloudFormation**.
132. Click the **`Create Stack`** button.
133. Give the Stack an appropriate name.
134. Click **Provide an S3 URL to template**, and type or paste the URL of the CloudFormation template: https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc/sys-ops-lab-vpc.template
135. Click the **`Next Step`** button to proceed.
136. On the **Specify Parameters** panel, most of the parameters needed are populated by the template. Using the information you recorded previously, do the following:
    f. In the **EnabledVPCAZ** field, type the name of 2 Availability Zones separated by a comma, for example: us-east-1a, us-east-1b. If you need the AZs in your account refer to the Name the Availability Zone section at the beginning of this lab.
    g. In the **KeyName** field, type the name of the qwiklab key pair.
137. Click the **`Next Step`** button to proceed.
138. (Optional) Add one or more tags.
139. Click **`Next Step`** to proceed.
140. Review the Stack Information and then click **`Create`** to create the stack.
141. Click the **`Events`** tab for your stack, if not already selected. Note the CloudFormation activities and statuses as the template is processed (it should take 5-10 minutes to fully provision the VPC and associated resources).

Conclusion

Congratulations! You now have successfully:

- Manually created a VPC
- Configured subnets in the VPC
- Created and attached an Internet Gateway (IGW)
- Defined routing tables and routes
- Setup security groups and rules to control access
- Deployed a NAT instance
- Deployed a VPC using a CloudFormation template

End Your Lab

142. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

143. Close any active SSH client sessions or remote desktop sessions.
144. Click the **End Lab** button on the *qwikLAB*™ lab details page.

145. When prompted for confirmation, click **OK**.
146. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operation on AWS
Linux Lab Guides

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
Amazon Elastic Compute Cloud (EC2)
Linux Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview
In this lab, you will build upon your previous lab experience with VPC, and deploy and bootstrap an instance within a VPC. For the most part, you will be using the AWS Command Line Interface (AWS CLI) to interact with EC2. This lab will help reinforce the understanding of EC2 relating to system usage, maintenance and management. You will learn techniques that can be helpful when creating *scripts* to automate a wide range of tasks. This is a fundamental lab, which other labs will build off of.

Objectives
This lab is intended to build off of a previous lab involving Amazon VPC and prepares you to further extend the infrastructure for deploying a web application

Technical Knowledge Prerequisites
No previous technical knowledge is strictly needed to go through this lab. However, to get the most out of it, it could be useful to have some basic knowledge on Linux servers operations. This includes being comfortable with the command line interface and the basic syntax of tools such as `ssh` and `curl`.

Topics Covered
The following topics will be covered:

- Creating Key Pairs
- Creating and configuring Security Groups
- Launching EC2 Linux Instances
- Describing and Tagging resources
- Connecting to EC2 Linux Instances

Amazon EC2 Instance Deployment

Process Overview
In this section, you will launch an EC2 Instance that will be a Database server. But before effectively launching the instance, a few steps are required.
All the steps and procedures described below could be done through the AWS Management Console. However, in this course, we focus on the AWS CLI. For this reason, we are going to do almost everything from the command line. Learning how to use the AWS CLI can be very helpful for you in the future, when you will want to write scripts to help you with management and automation of your own infrastructure.

The following steps describe the whole process:

1. Connect to the Bastion Host [AWS Management Console]
2. Inspect the Instance Metadata
3. Inspect the AWS CLI configuration and documentation [CLI]
4. Create a Security Group [CLI]
5. Create a Key Pair [CLI]
6. Launch the EC2 instance [CLI]
7. Connect to the instance and configure it [CLI]

Connect to the Bastion Host
All the steps prior to launching the EC2 Instance will be executed on the Bastion Host that you learned how to create on a previous lab. The benefits of using a Bastion Host include:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

- **Increased security**: by restricting access to your other resources to the Bastion Host only, there's less surface for attackers to hack into your infrastructure.
- **Auditing**: you could set up logging and auditing on the Bastion Host, so that every modification to your infrastructure would be securely logged and traceable. [*not covered on this lab*]
- **Easier to interact with the Services:** by using a Bastion Host for each infrastructure you manage on AWS, you don't need to rely on management software and settings being deployed on your own computer, on each of your company's computers, etc. All the tools are installed and configured in a central place – the Bastion Host – providing you with a homogeneous management environment that can easily be replicated.

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair
This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this manual called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group
It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

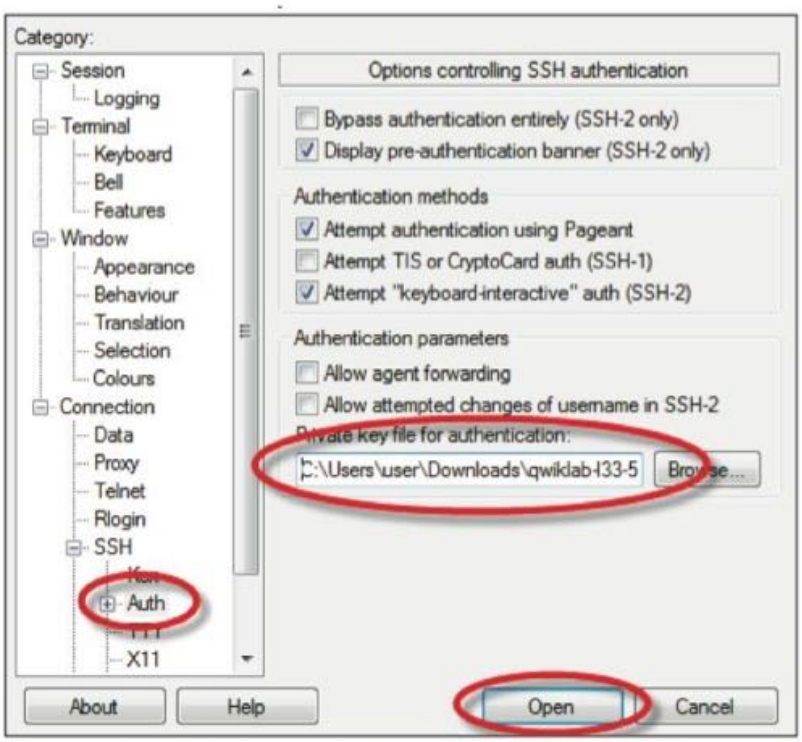
8. Retrieve your current public IP address by pointing your browser to http://ip.appspot.com
9. Open the **AWS Management Console**.
10. Select the **VPC** service.
11. From the menu on the left side, click **Security Groups**.
12. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion host"**.
13. On the bottom pane, click the **Inbound** tab.
14. From the **Create a new rule dropdown**, select SSH.
15. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
16. Click the **Add rule** button.
17. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host
18. Open the **AWS Management console**.
19. Select the **EC2** service.
20. From the menu on the left side, click **Instances**.
21. **Select the BastionHost instance** by clicking on the check box next to its name.
22. The Public IP is displayed on the bottom pane. Please write it down.

Connect to the Bastion Host
The steps to connect to the Bastion Host are different if you are using a Windows laptop or a Mac/Linux laptop. Please follow the appropriate set of instructions!
If you are using **Windows**:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

23. **Download PuTTY.exe** to a location of your choice (your Downloads folder, or your Desktop, for example) from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
24. **Double-click putty.exe** (the file you have just downloaded).
25. **Enter the Bastion Host's public IP into the Host Name input in Putty**. The public IP should have been obtained and written down in a previous section.
26. **Expand the SSH category** by double-clicking it. It is located on the menu on the left side.
27. Select the **Auth** sub-category (under the SSH category) by clicking it (not the + symbol next to it).
28. Click **Browser** and locate the PPK file (ending in **.ppk**). It is probably located in your Downloads folder.
29. Select the file and click **Open**.
30. In the **PuTTY** client, click **Open**.



31. When prompted, **click Yes** to allow PuTTY to cache the server's host key.

32. If you are prompted for a **username**, type **ec2-user**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

If you are using **Mac OS or Linux**:

33. Open the **Terminal** application.
34. Enter the commands below.
    In the **first command**, **change the path and file name** (~/Downloads/qwiklab-l33-5018.pem) to the location of the **.pem file** you downloaded earlier.
    In the **second command**, **change again the file name** and also **replace the string BASTION_IP** with the IP address of the Bastion Host you obtained earlier.

    chmod 400 ~/Downloads/qwiklab-l33-5018.pem
    ssh -i ~/Downloads/qwiklab-l33-5018.pem ec2-user@BASTION_IP

Inspect the Instance Metadata

For the next section, you will need some information about the Bastion Host (including the Region and the VPC it has been launched into and it's security group id). There are multiple ways to obtain it. In this lab, we are going to retrieve all that information by querying the Instance Metadata service.

Explore the Metadata service

Every EC2 instance has a Metadata service that can be queried by issuing HTTP GET requests to the endpoint http://169.254.169.254/. These requests are handled by the physical host's hypervisor. You are going to use the `curl` tool to explore the Metadata service

35. **Run the command**:

```
curl -s -w '\n' http://169.254.169.254/
```

36. **Observe** the list of versions of the service.
37. **Obtain information on the `latest` version**:

```
curl -s -w '\n' http://169.254.169.254/latest/
```

38. **Observe** the available categories of data, including `meta-data`.
39. **List** all the Metadata options available:

```
curl -s -w '\n' http://169.254.169.254/latest/meta-data/
```

40. **Inspect the `placement` related data**:

```
curl -s -w '\n' http://169.254.169.254/latest/meta-data/placement/
```

Retrieve the Region

41. **Retrieve the instance's availability zone**:

```
curl -s -w '\n' http://169.254.169.254/latest/meta-
data/placement/availability-zone
```

Note the Availability Zone from the previous command. If you discard the last letter, you obtain the name of the Region into which the Bastion Host has been launched.

42. **Take note** of the Region of the Bastion Host, by discarding the last letter of its Availability Zone.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Retrieve the VPC ID

43. **List** the instance's elastic network interfaces:

```
curl -s -w '\n' http://169.254.169.254/latest/meta-data/network/interfaces/macs/
```

44. **Save the MAC address** to an environment variable called BASTION_MAC:

```
export BASTION_MAC=$(curl -s -w '\n' http://169.254.169.254/latest/meta-data/network/interfaces/macs/)
```

45. **List** the available network information:

```
curl -s -w '\n' http://169.254.169.254/latest/meta-data/network/interfaces/macs/$BASTION_MAC/
```

46. **Save the VPC ID** to an environment variable called BASTION_VPC:

```
export BASTION_VPC=$(curl -s -w '\n' http://169.254.169.254/latest/meta-data/network/interfaces/macs/$BASTION_MAC/vpc-id)
```

Retrieve the Security Group ID

47. **Retrieve the ID of the Bastion Host's security group** :

```
curl -s -w '\n' http://169.254.169.254/latest/meta-data/network/interfaces/macs/$BASTION_MAC/security-group-ids
```

48. **Save the name of the Bastion Host's security group to an environment variable called BASTION_SG**:

```
export BASTION_SG=$(curl -s -w '\n' http://169.254.169.254/latest/meta-data/network/interfaces/macs/$BASTION_MAC/security-group-ids)
```

49. **Certify** that the environment variable has been correctly initialized:

```
echo $BASTION_SG
```

Retrieve the AMI ID

The Bastion Host has been launched with an Amazon Linux. During this lab, you are going to need the same AMI ID to launch a new instance. For this reason, you are going to retrieve the AMI ID from the Instance Metadata.

50. **Save the Bastion Host's AMI ID to an environment variable called AMI_ID**:

```
export AMI_ID=$(curl -s -w '\n' http://169.254.169.254/latest/meta-data/ami-id)
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Inspect the AWS CLI configuration and documentation

Change and inspect settings

Before you begin using the AWS CLI, you will change a setting that controls the format of the output of the commands. By default, the commands will output JSON, which is not the best format to interact with other Linux command line tools (such as grep, awk, sed, etc).

51. From the EC2 Instance prompt, **run the following command**:
    aws configure
52. Leave the Access Key blank and **hit enter**.
53. Leave the Secret Key blank and **hit enter**.
54. Specify the default region name as the **Bastion Host's region obtained earlier** and **hit enter**.
55. Specify the default output format: **text**, and **hit enter**.

The command will write the settings to the file located at `~/.aws/config`. If you wish to inspect the configuration file, follow the next steps:

56. **Run the following command**:

```
less ~/.aws/config
```

57. Scroll up and down **using the arrow keys** on your keyboard.
58. When you are done, return to the prompt by pressing the key **q** on your keyboard.

[optional] Obtaining help

The AWS CLI tools comes with extensive documentation. To read the manual, you just need to end any command with the argument `help`.

Let's explore the documentation with the following steps:

59. **Run the following command**:

```
aws help
```

60. Scroll up and down **using the arrow keys** on your keyboard.
61. When you are done, return to the prompt by pressing the key **q** on your keyboard.

The previous steps will show you general information on the AWS CLI, including a list of supported services. You can obtain help specific to each supported service and each supported API.

62. Obtain help on the EC2 service:

```
aws ec2 help
```

63. When you are done, return to the prompt by pressing the key **q** on your keyboard.
64. Obtain help on the RunInstances API of the EC2 service:

```
aws ec2 run-instances help
```

65. When you are done, return to the prompt by pressing the key **q** on your keyboard.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

[optional] Tip: auto-completion is your friend!
The AWS CLI comes with auto-completion feature enabled on the Bastion Host (also on any standard, recent Amazon Linux AMI). You invoke auto-complete with the TAB key. If more than one completion is available, you can press the TAB key twice to obtain a list of possible completions.

Try it now with the following steps:

66. **Type**:

```
aws[SPACE][TAB][TAB]
```

67. After seeing the list of possible completions for the services, **continue typing**:

```
aws ec2[SPACE][TAB][TAB]
```

When there are too many possible completions, it is likely that you'll be asked if you really want to list them all.

68. If you are prompted, "Display all XXX possibilities? (y or n)", hit **y**.
69. **Scroll down** by pressing the [SPACE] key, and terminate inspecting the possible completions by either reaching the end of the list or by hitting the key **q**.
70. **Continue typing**:

```
aws ec2 desc[TAB]
```

71. The AWS CLI will complete to *aws ec2 describe-*. **Continue typing**:

```
aws ec2 describe-[TAB][TAB]
```

72. Inspect the list through the end or hit **q** to interrupt. Continue typing:

```
aws ec2 describe-reg[TAB]
```

73. It will now fully complete the command **aws ec2 describe-regions**. **Hit enter** to obtain a list of the EC2 regions.

Getting used to the auto-completion feature and to reading the help information can help you increase your productivity.

Create a Security Group

As you should know by now, it is recommended to have a separate Security Group for each logical grouping of servers you maintain (for example, one for the web-servers, another one for the application servers, another one for a specific background worker, another one for your database layer, and so on).

For this reason, you are going to create a specific Security Group for the Database server you are going to launch. The name of the security group will be *db-sg*. For now, it will allow ingress SSH traffic from the Bastion Host only, so that you will be able to connect to it and configure it.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

74. **Create the security group by running the following command, and save its ID to an environment variable called DB_SG**:

```
export DB_SG=$(aws ec2 create-security-group --group-name db-sg --
vpc-id $BASTION_VPC --description database-security-group --query
GroupId)
```

75. **Authorize ingress SSH from the Bastion Host security group**:

```
aws ec2 authorize-security-group-ingress --group-id $DB_SG --
protocol tcp --port 22 --source-group $BASTION_SG
```

Don't forget to use auto-completion to save you time and avoid mistakes!

Create a Key Pair

It is a security best practice to have a different Key Pair for each logical grouping of servers you maintain. For this reason, you are now going to create a Key Pair for your Database server(s).

[optional] Pipes (i.e., "|") and *tee*

If you already know the basics of piping commands on Linux and the *tee* command, please skip directly to the next steps.

A very powerful feature of the Linux command line is known as *pipes*. A pipe allows you to use the output of one command as the input of another command. You are now going to use an AWS CLI command that creates a Key Pair and outputs the Key Pair data to the screen. However, outputting to the screen isn't really useful, because you need the Key Pair data in a file to use it.

Some commands output to screen pieces of data that you would need to save into a file. To avoid *copy-and-paste*, there's a tool called *tee*. Simplifying, all this tool does is take its input, save into a file you name, and also output the data to the screen. Think of a T-junction with one input and two outputs (the file and the screen).

By *piping a command through tee*, you effectively save the command's output to a file and you also receive the output in your screen, so that you can inspect it.

Create the Key Pair and set key file permissions

76. **Run the following command**:

```
aws ec2 create-key-pair --key-name db-kp --query KeyMaterial | tee
~/db-kp.pem
```

77. **Change the Key Pair file POSIX permissions**:

```
chmod 400 ~/db-kp.pem
```

It is very common to forget to set the appropriate POSIX permissions on the Key Pair file. If you forget to do it, you'll receive an error message from OpenSSH when you try to use the key to connect to a server.

Launch the EC2 instance

The last piece of information needed to launch the EC2 instance is the Private Subnet id. After obtaining this information, you are going to launch the instance and save its ID to an environment variable called DB_INSTANCE_ID which will be used later.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Retrieve the Private Subnet id

When the Lab has been created, all the resources have been properly tagged. For example, the Subnets the tags Network=Public and Network=Private. We will exploit these tags to retrieve the Private Subnet ID. This should show you the importance of properly tagging your resources!

78. **Save the Private Subnet ID to an environment variable called PRIV_SUBNET**:

```
export PRIV_SUBNET=$(aws ec2 describe-subnets --filters
'Name=tag:Network,Values=Private' --query 'Subnets[*].SubnetId')
```

Launching the EC2 instance

79. **Create the instance**:

```
export DB_INSTANCE_ID=$(aws ec2 run-instances --image-id $AMI_ID --
key-name db-kp --security-group-ids $DB_SG --instance-type m1.small
--subnet-id $PRIV_SUBNET --query 'Instances[*].InstanceId')
```

80. **Tag the instance**:

```
aws ec2 create-tags --resources $DB_INSTANCE_ID --tags
'Key=Name,Value=DB1'
```

Verifying the EC2 instance state

81. **Run the following command**:

```
aws ec2 describe-instances --instance-ids $DB_INSTANCE_ID
```

As you can see, there is a lot of information you can obtain from this command. You can use the --query option to retrieve only the state:

82. **Run the following command**:

```
aws ec2 describe-instances --instance-ids $DB_INSTANCE_ID --query
'Reservations[*].Instances[*].State.Name'
```

83. **Repeat the previous command** until the instance is in the *running* state.

This is particularly useful when you are creating a script that must wait for an instance to be in the RUNNING state before it can proceed. Just have the script run the previous command (with the --query) repeatedly until a condition is met, then your script can proceed.

Connect to the instance and configure it

This is the time to connect to your newly launched instance and make a Database server of it.
First, you need to obtain the IP of the newly launched instance. After that, it's just a matter of using OpenSSH installed on the Bastion Host to connect to the Database server instance.

Obtain the DB instance's IP

84. **Retrieve the DB instance's IP**:

```
export DB_INSTANCE_IP=$(aws ec2 describe-instances --instance-ids
$DB_INSTANCE_ID --query
'Reservations[*].Instances[*].PrivateIpAddress')
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

85. If you are curious, run the following command to see the hostname:

```
echo $DB_INSTANCE_IP
```

Connect to the DB instance
To connect to the instance, you need the following information:

- The instance's IP
- The instance's default user name
- The instance's key pair file

The IP has just been retrieved on the previous step. The default user name is defined by the AMI – since we used an Amazon Linux, this value is *ec2-user*. Finally, the key file has been generated previously.

The general syntax of OpenSSH you will use is: ssh -i /path/to/key/file.pem username@hostname.example.com.

86. **Connect to the DB instance**:

```
ssh -i ~/db-kp.pem ec2-user@$DB_INSTANCE_IP
```

You should see the prompt of the DB instance! In the next sections, you are going to install a Database server and make sure it starts automatically on boot, should your instance restart for any reason.

Comparison with AWS Management Console

You have just created all your infrastructure using the AWS CLI. Whether you use the CLI, the SDKs or the Management Console, you are interacting with the same sources and managing the same resources. In this part, you are going to browse the AWS Management Console to locate all the resources you created using the AWS CLI.

Locate the Key Pair

87. Open the **AWS Management Console**.
88. Select the **EC2** service.
89. From the menu on the left side, click **Key Pairs**. Note: you might need to scroll down the left side menu.
90. **Verify** that the key pair you created (*db-kp*) is shown on the list.

Locate the Security Group

91. From the menu on the left side, click **Security Groups**. Note: you might need to scroll down the left side menu.
92. **Verify** that the security group you created (*db-sg*) is shown on the list.
93. **Select the security group db-sg**.
94. **Click the Inbound tab**.
95. **Verify** that the port 22 is open to the Bastion Host's security group.

Locate the Custom AMI

96. From the menu on the left side, click **AMIs**.
97. **Verify** that there is a custom AMI on the list.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Locate the Instances

98. From the menu on the left side, click **Instances**.
99. **Verify** that there is one instance called **DB1** and one instance called **DB2**.

Conclusion

In this lab, you've learned how to set up the AWS Command Line Interface on a Bastion Host and how to browse its documentation. You've also used the AWS CLI to create some EC2-related resources, including Key Pairs, Security Groups and Instances. Also, very important, you learned how to tag resources and how to use the tags to query and locate the resources you need.

Most importantly, you've learned some important techniques to interact with the services programmatically. You are now able to create scripts that will create resources and monitor they state.

In the next labs, you are going to learn how to interact with the other services to create a complete infrastructure in AWS.

End Your Lab

100. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

101. Close any active SSH client sessions or remote desktop sessions.
102. Click the **End Lab** button on the *qwikLAB*™ lab details page.

103. When prompted for confirmation, click **OK**.
104. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS Amazon Elastic Block Store (EBS) Linux Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

In this lab, you will build upon your previous lab experience with EC2, and learn about EBS. You will have the opportunity to work with volumes and the manipulation thereof, provisioned IOPS, snapshots, and using the AWS Management Console and AWS CLI. This lab will help reinforce the understanding of EC2 relating to storage, maintenance, and management.

Objectives

This lab is intended to build off of a previous lab involving Amazon EC2 and prepares you to further extend your knowledge of storage solutions.

Note: All text in ORANGE requires you to replace it with data from your lab environment.
Note: You're expected to see JSON output when you run the provided commands, as JSON is the default output. You may find that running these commands from a PowerShell session on Windows (to take advantage of "`ConvertFrom-Json`") or with the aid of `jq` on Linux improves legibility and navigation.
Note: We have created a bastion host instance for you to use in this lab. Under normal circumstances, this is created by the customer, and is a best practice.
Note: Generated on the bastion host will be 2 files: aws_cli_config and awsstudent.id.rsa. The awsstudent.id.rsa is what you will use to connect to and create your instances with.

Topics Covered

The following topics will be covered:

- Using the Amazon AWS CLI
- Executing Amazon EC2 commands
- Connecting to Amazon EC2 instances
- Creating, configuring and attaching volumes
- Creating and instantiating snapshots
- Creating volumes from snapshots
- Creating provisioned IOPS volumes
- Striping volumes

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Connect to the Bastion Host

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair

This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this manual called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group

It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

1. Retrieve your current public IP address by pointing your browser to http://ip.appspot.com
2. Open the **AWS Management Console**.
3. Select the **VPC** service.
4. From the menu on the left side, click **Security Groups**.
5. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion host"**.
6. On the bottom pane, click the **Inbound** tab.
7. From the **Create a new rule dropdown**, select SSH.
8. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
9. Click the **Add rule** button.
10. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host

11. Open the **AWS Management console**.
12. Select the **EC2** service.
13. From the menu on the left side, click **Instances**.
14. **Select the BastionHost instance** by clicking on the check box next to its name.
15. The Public IP is displayed on the bottom pane. Please write it down.

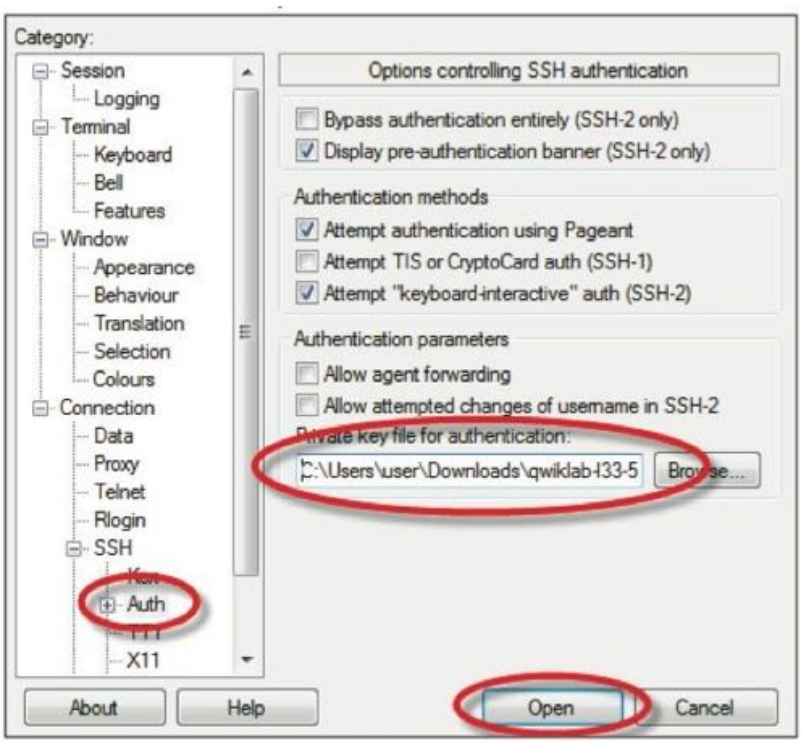
Connect to the Bastion Host

The steps to connect to the Bastion Host are different if you are using a Windows laptop or a Mac/Linux laptop. Please follow the appropriate set of instructions!
If you are using **Windows**:

16. **Download PuTTY.exe** to a location of your choice (your Downloads folder, or your Desktop, for example) from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
17. **Double-click putty.exe** (the file you have just downloaded).
18. **Enter the Bastion Host's public IP into the Host Name input in Putty**. The public IP should have been obtained and written down in a previous section.
19. **Expand the SSH category** by double-clicking it. It is located on the menu on the left side.
20. Select the **Auth** sub-category (under the SSH category) by clicking it (not the + symbol next to it).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

21. Click **Browser** and locate the PPK file (ending in **.ppk**). It is probably located in your Downloads folder.
22. Select the file and click **Open**.
23. In the **PuTTY** client, click **Open**.



24. When prompted, **click Yes** to allow PuTTY to cache the server's host key.

25. If you are prompted for a **username**, type **ec2-user**

If you are using **Mac OS or Linux**:

26. Open the **Terminal** application.
27. Enter the commands below.
    In the **first command**, **change the path and file name** (~/Downloads/qwiklab-l33-5018.pem) to the location of the **.pem file** you downloaded earlier.
    In the **second command**, **change again the file name** and also **replace the string BASTION_IP** with the IP address of the Bastion Host you obtained earlier.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
chmod 400 ~/Downloads/qwiklab-l33-5018.pem
ssh -i ~/Downloads/qwiklab-l33-5018.pem ec2-user@BASTION_IP
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

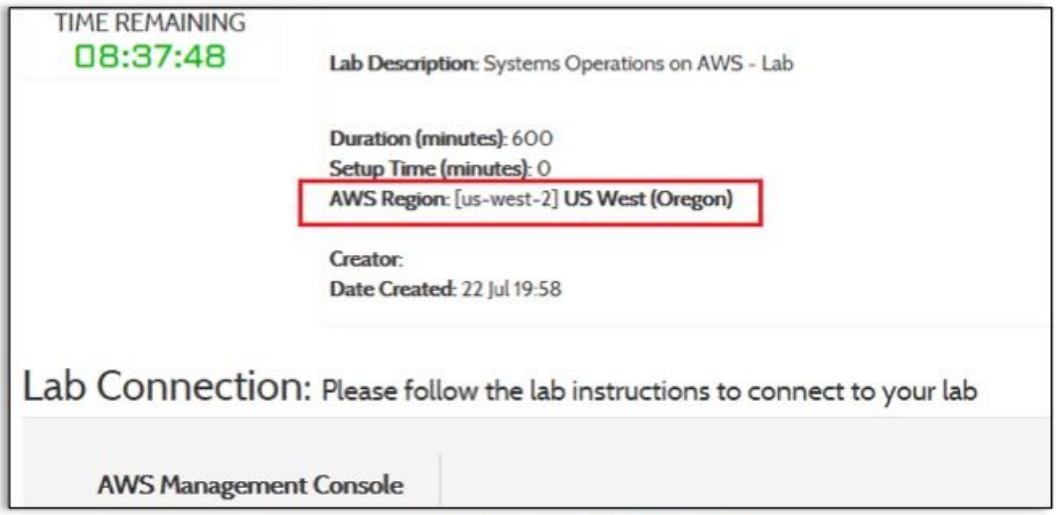
Using AWS EC2 CLI

For the rest of the lab, you will be running AWS CLI commands from the Bastion host against your storage host. Unless otherwise noted, the following sections must be run from the bastion host provided within your qwikLAB. In order to use the AWS CLI you would generally need to install the tools, and configure the instance with your account details. For this lab, the tools are preconfigured for you. For your information and further reading, please go to http://aws.amazon.com/cli.

PEM Configuration and AWS CLI from Linux Command Prompt

In order to connect to the bastion instance, you will need to configure the qwikLAB pem file on the bastion host. Additionally further create a configuration file that the AWS CLI will have its settings in.

1. If your lab has already been started, you can verify your region here:



We will refer to this value throughout the Module as **xy-region-1**

2. From your qwikLAB page, download your PEM

3. Get the IP of your ***bastion host*** (not the newly created database EC2 Instance), make note of the EIP
(externally visible Elastic IP address) and put that value into your SSH client.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: Your *username* on Amazon Linux machines will be **ec2-user** (NOT root).

4. You should already be in your home directory. If not, change directory to the ec2-user home directory.

```
cd
```

5. Without qwikLAB, you would normally create your own configuration file and use an AWS environment variable to refer to it (e.g. `AWS_CONFIG_FILE=/home/ec2-user/aws_cli_config`). In qwikLAB environments, the configuration file is created for you already. Here is a sample of commented configurable fields you can specify to get the AWS CLI to behave differently.

```
[default]
#aws_access_key_id=AKIAYourAccessKey
#aws_secret_access_key=ASDFyourSecretAccess
region=xy-region-1
#output=text
```

Your lab machine already have this file created with the appropriate region populated.

- Typically we would not comment-out the `aws_access_key_id` or the `aws_secret_access_key`, however this lab relies on an IAM role for identity so one does not need to be specified here. It is shown in this context for additional clarity.
- The `output=text` option is commented out because we will use JSON in this lab. However, the tabular output option is available and shown here for clarity.

6. Show the contents of your `aws_cli_config` file with the following command

```
cat aws_cli_config
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

For the purposes of this lab, it only contains region, but outside of qwikLAB you will need to create one similar to the file mentioned previously

7. You can test that everything worked by getting a global list of available EC2 regions. You can get that list by running the following command:

```
aws ec2 describe-regions
```

Working with Volumes

Boot Volumes in Linux

This lab assumes you have the instance ID of your EC2 instance readily available. For the purposes of this Module, you will substitute *i-YourInstanceId* with the instance ID of the EC2 storage instance created for you. You will log in to your bastion host, and run the following commands against your new EC2 Instance that you created previously.

Getting Instance IDs

8. Retrieve the Storage Server instance id and set it to a local variable:

```
export STORAGE_SERVER_ID=$(aws ec2 describe-instances --filter
Name=tag:Name,Values=STORAGE --query
"Reservations[*].Instances[*].InstanceId" --output text)
```

Like the `ec2 describe-instances` command set we used in the EC2 lab, note the difference in the query string.

Resizing Boot Volumes

9. Notice there is an attached volume ID on your EC2 instance:

```
aws ec2 describe-instances --instance-ids $STORAGE_SERVER_ID
```

Take note of your availability zone (AZ). It will look similar to the region you took note of earlier. All of the resources we use for the rest of these Modules, unless otherwise noted, must be located within the same AZ. For the purposes of this lab, we will generally refer to it as, *az-asdf-1a*, which you will substitute with the real value in the following steps

If the output is too verbose for you to find the AZ, you can filter it by running the following:

```
aws ec2 describe-instances --instance-ids $STORAGE_SERVER_ID | grep
Zone
```

Take note of your volume ID, which will be referred to as *vol-TheSmallDefaultVolume*. We will resize this to a newer, larger volume, while retaining all of the same information already on the

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

disk. Your volume ID should look similar to this:

```
                "Hypervisor": "xen",
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/sda1",
                        "Ebs": {
                            "Status": "attached",
                            "DeleteOnTermination": true,
                            "VolumeId": "vol-3e5b4c78",
                            "AttachTime": "2013-04-09T21:57:01.000Z"
                        }
                    }
                ],
                "Architecture": "x86_64",
```

Notice the volume is attached and the size is 8GB:

```
aws ec2 describe-volumes --volume-ids vol-TheSmallDefaultVolume
```

10. Initiate a snapshot of the 8GB volume

```
aws ec2 create-snapshot --volume-id vol-TheSmallDefaultVolume
```

Notice that the snapshot will have a unique ID. ***Make a note of the snapshot ID***. We will refer to it generally as *snap-YourVolumesSnapshot*, which you will substitute with the real value in later steps. Also note that the EC2 instance remains available during this process.

11. Watch the snapshot progress using this command:

```
aws ec2 describe-snapshots --snapshot-ids snap-YourVolumesSnapshot
```

Notice the Snapshots:State will show pending while the snapshot is in progress. As you wait for the snapshot to complete, you can re-run this command. When finished, the Snapshots:State will show as completed, and Progress 100%.

12. Create a larger 10GB volume based on the snapshot. Your snapshot, volume, and EC2 instance must all reside in the same availability zone. You must use the same AZ of EC2 instance noted previously, which should also match the AZ of your bastion host. Look it up on the EC2 Instance section of the AWS Management Console if you have not already made a note of the AZ

```
aws ec2 create-volume --snapshot-id snap-YourVolumesSnapshot --size 10
--availability-zone az-asdf-1a
```

Note that the new volume will have its own volume ID returned by this command. We will refer that volume ID as vol-TheLargerVolume, which you will substitute with the real value in the following steps

13. Run/re-run this command as needed to watch progress

```
aws ec2 describe-volumes --volume-ids vol-TheLargerVolume
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Notice the State field is pending. When the new, larger volume is done being created, it will show as available.

14. Now, detach the old small volume and attach the new bigger volume. Stop the instance:

```
aws ec2 stop-instances --instance-id $STORAGE_SERVER_ID
```

15. Then detach the old volume:

```
aws ec2 detach-volume --volume-id vol-TheSmallDefaultVolume
```

16. Make sure the small volume state changes to *available* to ensure detaching completed, by running the following command:

```
aws ec2 describe-volumes --volume-ids vol-TheSmallDefaultVolume
```

17. When the small volume has successfully detached you may attach the larger volume, with this command:

```
aws ec2 attach-volume --volume-id vol-TheLargerVolume --instance-id
$STORAGE_SERVER_ID --device /dev/sda1
```

18. Watch status of the attach with a describe command:

```
aws ec2 describe-volumes --volume-ids vol-TheLargerVolume
```

19. When the larger volume has successfully attached, the state will change to *in-use*. Then you can start the EC2 instance and it will use the new, larger volume with this command:

```
aws ec2 start-instances --instance-id $STORAGE_SERVER_ID
```

20. Watch the new instance enter a running state, look for "running" ensuring it is active from the previous command:

```
aws ec2 describe-instances --instance-ids $STORAGE_SERVER_ID --query
"Reservations[*].Instances[*].State.Name"
```

21. Notice the old 8GB volume is no longer connected to anything and the state is *available*.

```
aws ec2 describe-volumes --volume-ids vol-TheSmallDefaultVolume
```

22. Notice again that the new 10GB volume is attached to our EC2 instance and the state is `in-use`.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
aws ec2 describe-volumes --volume-ids vol-TheLargerVolume
```

23. Retrieve the `PublicIp` of your new Storage Server. We will refer to this value as `StorageHostIp`.

```
aws ec2 describe-instances --instance-id $STORAGE_SERVER_ID --query
"Reservations[*].Instances[*].NetworkInterfaces[*].Association.PublicI
p"
```

24. SSH from your host into the EC2 instance (storage server) created for you.

```
ssh -i YourFile.pem ec2-user@StorageHostIp
```

When prompted:
`Are you sure you want to continue connecting (yes/no)?`
respond with `yes`

25. Use the command *df -h* to view the volume information in the storage instance. Notice it is not using all of the available disk space yet.

```
df -h
```

26. Tell the OS to incorporate the newly available disk space with the *resize2fs* command.

```
sudo resize2fs /dev/sda1
```

Note: The command will confirm that the file system length has changed
`The filesystem on /dev/sda1 is now 2621440 blocks long.`

27. Use *df -h* again to see that the OS recognizes (and can use) the newly available disk space

```
df -h
```

28. Return to your original console on the bastion host, and safely delete the old 8GB volume:

```
aws ec2 delete-volume --volume-id vol-TheSmallDefaultVolume
```

29. And delete the intermediate snapshot:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
aws ec2 delete-snapshot --snapshot-id snap-YourVolumesSnapshot
```

Working with Ancillary Volumes

30. Before we attach a drive to an instance, we will stop the instance first. Run the following command to stop your new storage server:

```
aws ec2 stop-instances --instance-id $STORAGE_SERVER_ID
```

31. Create a PIOPs volume, and attach it to your new database server by running the following command:

```
aws ec2 create-volume --availability-zone az-asdf-1a --size 20 \
--volume-type io1 --iops 600
```

Note the volume ID of this fast disk drive, which we will refer to vol-FastDrive.

32. Watch the drive until is has been created and is available:

```
aws ec2 describe-volumes --volume-id vol-FastDrive
```

33. Attach the new fast drive to your new database server using the following command:

```
aws ec2 attach-volume --volume-id vol-FastDrive --instance-id
$STORAGE_SERVER_ID --device /dev/sdc
```

Striping Volumes

Next you will create volumes and stripe them. This is an alternate way to get increased IOPs, however it is not a fault tolerant solution. It may be useful for temporary processing where failure can be handled (e.g. processing jobs on data where the job can start over if data is lost). Note their IDs so you can watch their progress. For this exercise we will refer to them as vol-stripe**1**, vol-stripe**2**, vol-stripe**3**, vol-stripe**4**.

34. First, create 4 volumes by repeating the following command:

```
aws ec2 create-volume --availability-zone az-asdf-1a --size 5
```

35. Watch the creation status of the two new volumes until they become available:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
aws ec2 describe-volumes --volume-ids vol-stripe1 vol-stripe2 \
vol-stripe3 vol-stripe4
```

36. Attach the newly created volumes to your new server by repeating the following command:

```
aws ec2 attach-volume --volume-id vol-stripe1 --instance-id
$STORAGE_SERVER_ID --device /dev/sdd
```

37. Repeat the previous command, using the following mapping:

volume-id	device
vol-stripe2	/dev/sde
vol-stripe3	/dev/sdf
vol-stripe4	/dev/sdg

Striping Provisioned IOPS Volumes

Next you will create volumes with provisioned IOPS and stripe them. This is an alternate way to get increased IOPs, however it is not a fault tolerant solution. It may be useful for temporary processing where failure can be handled (e.g. processing jobs on data where the job can start over if data is lost). Note their IDs so you can watch their progress. For this exercise we will refer to them as vol-pstripe1, vol-pstripe2, vol-pstripe3, vol-pstripe4.

38. First, create 4 volumes by repeating the following command:

```
aws ec2 create-volume --availability-zone az-asdf-1a --size 10 \
--volume-type io1 --iops 200
```

39. Watch the creation status of the two new volumes until they become available:

```
aws ec2 describe-volumes --volume-ids vol-pstripe1 vol-pstripe2 vol-
pstripe3
vol-stripe4
```

40. Attach the newly created volumes to your new server by repeating the following command:

```
aws ec2 attach-volume --volume-id vol-pstripe1 --instance-id
$STORAGE_SERVER_ID --device /dev/sdh
```

41. Repeat the previous command, using the following mapping:

volume-id	device
vol-pstripe2	/dev/sdi

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
vol-pstripe3        /dev/sdj
vol-pstripe4        /dev/sdk
```

Pre-warming Volumes

Before using the storage volumes, it is considered a best practice to pre-warm your volume. To do this we have provided a tool on the storage server called `prewarm.py` to do this for you. Normally, you would need to run a command like `dd` on the entire volume to complete the pre-warming process.

42. With the new drives attached, you can start your new storage server using the following command:

```
aws ec2 start-instances --instance-id $STORAGE_SERVER_ID
```

43. Like before, SSH from your host to the provided storage server:

```
ssh -i YourFile.pem ec2-user@StorageHostIp
```

44. Pre-warm the single PIOPS volume by running the following command:

```
sudo prewarm.py --dev /dev/sdb
```

45. Get the OS to recognize your PIOPs volume by running the following commands:

```
sudo mkdir /mnt/piops1
```

```
sudo mkfs.ext3 /dev/sdb
```

```
sudo mount /dev/sdb /mnt/piops1
```

46. Get the OS to pre-warm, stripe, and then recognize your additional volumes with this series of commands:

```
sudo mkdir /mnt/stripe1 /mnt/stripe2
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
for dev in /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg /dev/sdh\
/dev/sdi /dev/sdj /dev/sdk; do sudo ./prewarm.py --dev ${dev}; done;
```

```
sudo vgcreate vg1 /dev/sdd /dev/sde /dev/sdf /dev/sdg

sudo vgcreate vg2 /dev/sdh /dev/sdi /dev/sdj /dev/sdk
```

```
sudo lvcreate -i2 --size 10G -n lv1 vg1

sudo lvcreate -i2 --size 10G -n lv2 vg2
```

The parameter `-i` in the `lvcreate` command specifies the number of physical volumes to stripe across. If you wanted to stripe across more than 2 drives, you would have to specify that number.

```
sudo mkfs.ext3 /dev/vg1/lv1

sudo mkfs.ext3 /dev/vg2/lv2
```

```
sudo mount /dev/vg1/lv1 /mnt/stripe1

sudo mount /dev/vg2/lv2 /mnt/stripe2
```

47. Next, update /etc/fstab to ensure these mounted drives are automatically mounted on reboot.

```
sudo vi /etc/fstab
```

48. Type `i` to put vi into insert mode, then append the following lines

```
/dev/sdb     /mnt/piops1   ext3     defaults       0     2
/dev/vg1/lv1 /mnt/stripe1  ext3     defaults       0     2
/dev/vg2/lv2 /mnt/stripe2  ext3     defaults       0     2
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Your fstab should look like this:

```
#
LABEL=/         /               ext4    defaults, noatime     1 1
tmpfs           /dev/shm        tempfs  defaults              0 0
devpts          /dev/pts        devpts  gid=5, mode=620       0 0
sysfs           /sys            sysfs   defaults              0 0
proc            /proc           proc    defaults              0 0
/dev/sdb        /mnt/piops1     ext3    defaults              0 2
/dev/vg1/lv1    /mnt/stripe1    ext3    defaults              0 2
/dev/vg2/lv2    /mnt/stripe2    ext3    defaults              0 2
~
~
~
~
~
~
~
~
-- INSERT --
```

49. Press `ESC` to get vi back into visual mode, and then type `:wq<enter>` to save your changes.

50. You can now verify that your new volumes are available by running the following command:

```
df -h
```

Comparing Standard IOPS, Provisioned IOPS, and Combined Provisioned IOPS

For this next section we're going to be using the flexible I/O tester *fio* to put the volumes through their paces and test their throughput and I/Os per second.

51. Run the following command to perform random read/writes on a 512MB file object and replace `target-mount` from the adjoined table:

```
fio --directory=target-mount --rw=randrw --bs=512 --size=512m --
runtime=90 \
--name=filerw
```

Target Volume Type	**`target-mount`**
Standard IOPS	/
Provisioned IOPS	/mnt/piops
Striped Volume (Standard IOPS)	/mnt/stripe1
Striped Volume (Provisioned IOPS)	/mnt/stripe2

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

While running the command for each observe the IOPS performance section during the progress for each type, see the following example.

```
[root@ip-10-0-0-174 ~]# fio --directory=/mnt/piops --rw=randrw --
size=512m --bs=512 --runtime=90 --name=filerandwrite

filerandwrite: (g=0): rw=randrw, bs=512-512/512-512/512-512,
ioengine=sync, iodepth=1
fio-2.0.13
Starting 1 process
Jobs: 1 (f=1): [m] [11.0% done] [95K/88K/0K /s] [132 /156 /0  iops]
[eta 01m:05s]
```

Conclusion

Congratulations! You have successfully:

- Deployed an Amazon AWS AMI
- Used the Amazon AWS CLI to execute commands against remote instances
- Used the Amazon AWS CLI to view the state of operations in AWS
- Connected to Amazon EC2 instances from a bastion host
- Created, configured and attached volumes to instances
- Created and modified EBS volumes
- Created PIOPs volumes
- Striped volumes
- Compared the performance for PIOPS vs. Standard vs. Combined PIOPS vs. Combined Standard

End Your Lab

53. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

54. Close any active SSH client sessions or remote desktop sessions.
55. Click the **End Lab** button on the *qwikLAB*™ lab details page.

56. When prompted for confirmation, click **OK**.
57. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS Tagging Linux Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

Tags are an essential part of keeping a large AWS footprint organized. Tags can be used to sort AWS resources, and can also be used to search for particular resources. Although rather simple in and of themselves, tags are one of the fundamental building blocks of effectively managing a dynamic infrastructure.

In this lab you will:

- Sort instances by tag using the AWS Management Console
- Search for tags using the AWS Management Console
- Search for tags using the Amazon EC2 CLI
- Create custom tags using the AWS Management Console
- Create custom tags using the Amazon EC2 CLI

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with the Linux shell and using ssh to access Linux servers.

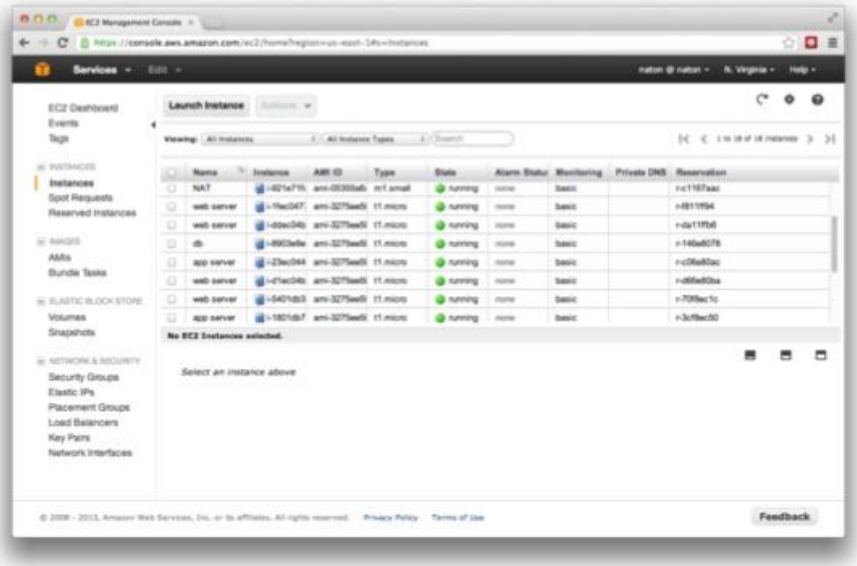
Topics Covered

This lab will focus entirely tagging in the context of EC2 instance; tags for other AWS resources utilize the same concepts, but will not be introduced in this lab.

Working with tags in the console

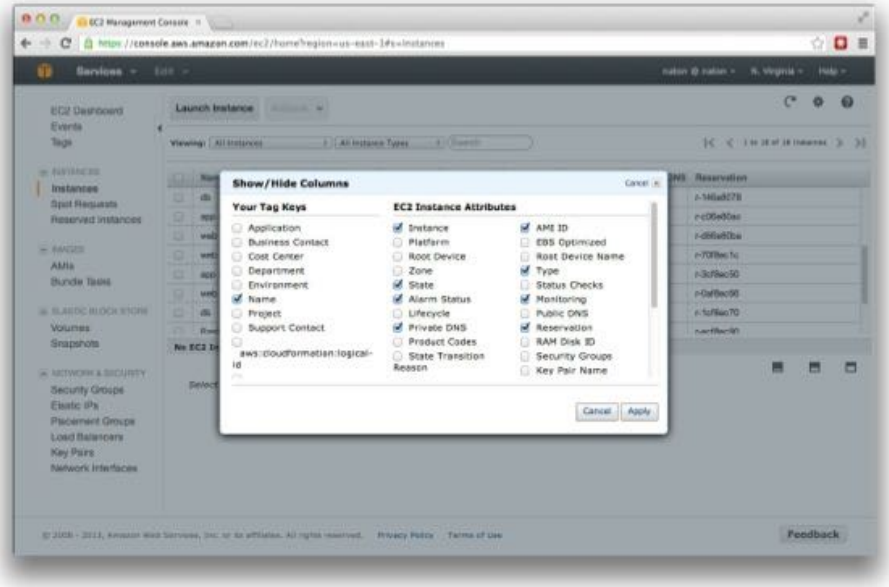
Sorting and searching by tags

1. Log into the AWS Management Console and navigate to **EC2->Instances->Instances** (ensure that the correct region is selected). There should be at least 10 running instances visible.



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

2. Click on the Settings button



3. Under the heading "Your Tag Keys", select the checkboxes for "Application", "Name", and "Environment". Click "Apply".

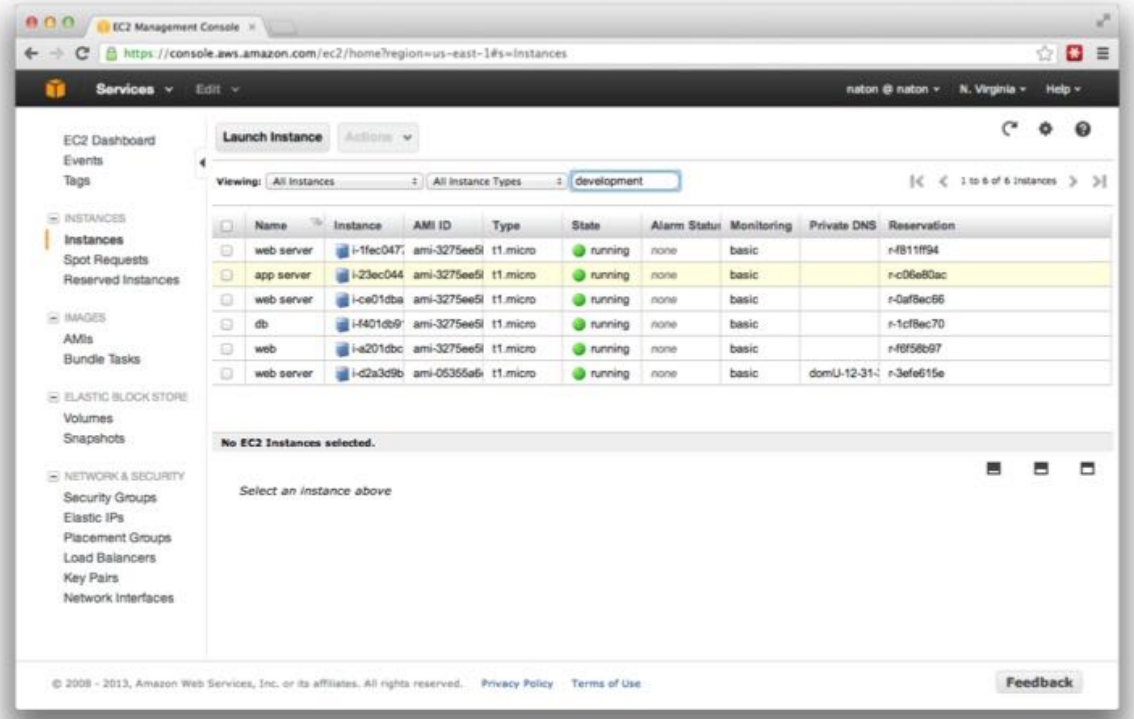
Sort

4. Click on the "Application" column to sort all instance by application. Note how many servers are dedicated to each application.

Search

5. In the search bar, type "development" to only show instances with "development" in their tag values.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



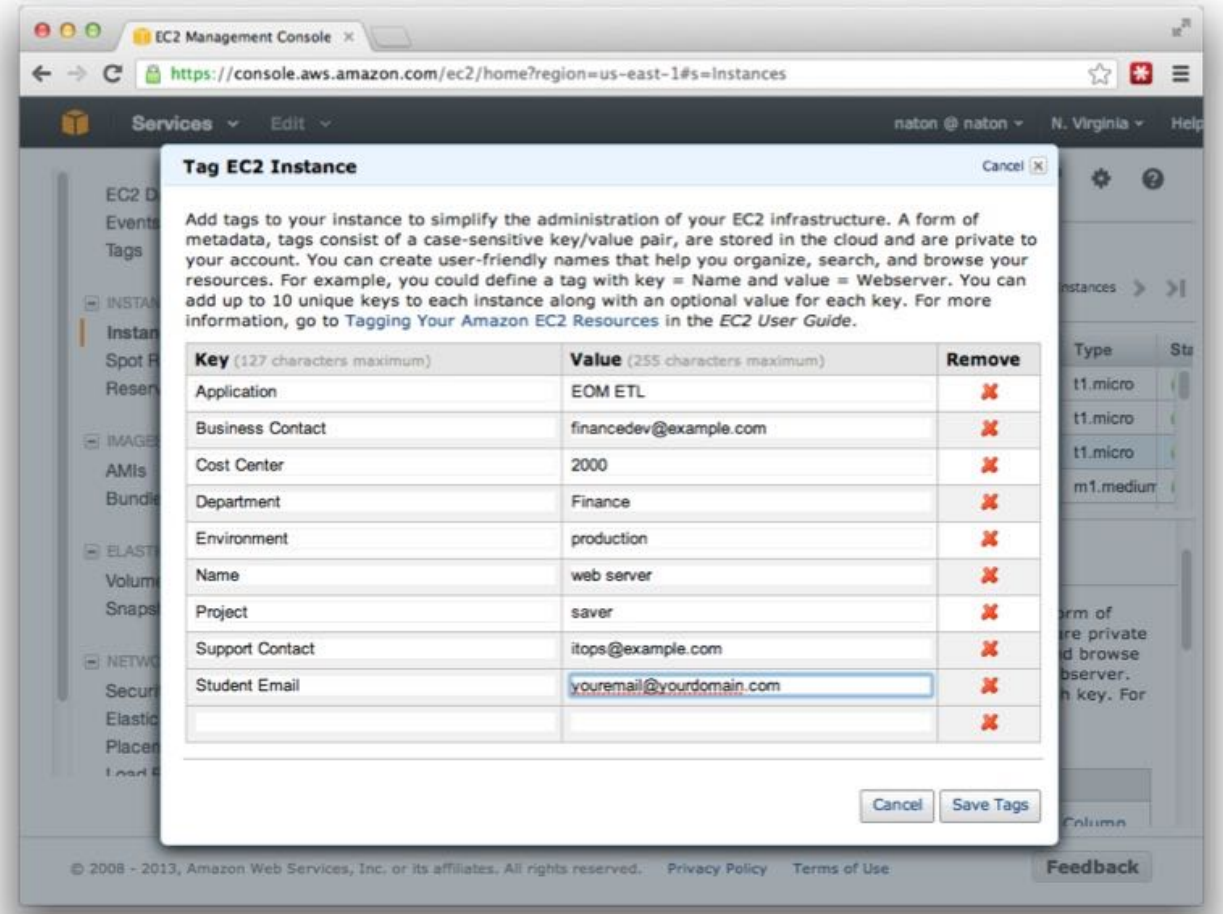
Get / change instance tags

6. Click on the checkbox for any one (but only one) of the instances.

7. In the bottom half of the screen, navigate to the "Tags" tab and scroll through the tags. Notice the tags at the bottom that start with "aws:" – these are immutable and cannot be deleted.

8. Click on the "Add/Edit Tags" button, and add a tag with key "Student Email", and your email address as the value. Click on "Save Tags" to apply.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



Working with the tagging CLI

9. Using your preferred SSH client, log into the BASTION Linux host

10. Run the command to show all tags:

```
aws ec2 describe-tags
```

This will show every tag for all EC2 resources. This generally isn't very useful, but could be used as the basis for a script to gather information, search, or take other action.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

11. Search explicitly for the tags pertaining to the instance that was just edited in the AWS Management Console:

```
aws ec2 describe-tags --filters "Name=resource-id,Values=i-
01234567"
```

[note:in all these examples, you will need to replace **i-01234567** with the correct instance id]

12. Alternately, try searching for any tags with the key "Student Email":

```
aws ec2 describe-tags --filters "Name=key,Values=Student Email"
```

Note that this only comes up with the tag that was added from the console.

13. You can add or modify tags using the **aws ec2 create-tags** command. Try changing the number of the Cost Center in the instance you just modified:

```
aws ec2 create-tags --resources i-01234567 --tags "Key=Cost
Center,Value=2001"
```

14. Refresh the console to reflect the change.
15. Finally, delete the custom tag created previously:

```
aws ec2 delete-tags --resources i-01234567 --tags "Key=Student
Email"
```

Bonus activities

- Find all instances associated with Cost Center 1000
- Change all tags of "Cost Center=1000" to "Cost Center=1001"

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Bonus answers:

1) Find all instances associated with Cost Center 1000

```
aws ec2 describe-tags --filters Name=key,Values="Cost
Center",Name=value,Values=1000
```

2) Change all tags of "Cost Center=1000" to "Cost Center=1001"
This can be done via a variety of methods. Here's one using just shell commands.

Assuming that the previous command yields a series of lines like:

```
instance    i-01234567  1000  Cost Center
```

We can pipe it through awk to extract all the instances, then pipe that through "**aws ec2 create-tags**":

```
aws ec2 describe-tags --filters Name=key,Values="Cost
Center",Name=value,Values=1000 --output text | awk '{print $4}' |
while read instance ; do aws ec2 create-tags --resources $instance
--tags Key="Cost Center",Value=1001 ; done
```

Conclusion

Congratulations! You now have successfully:

- Used the console to sort and filter instances by tags
- Used the console to manage and change tags
- Used the CLI to list and search by tags
- Used the CLI to edit and delete tags

End Your Lab

58. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

59. Close any active SSH client sessions or remote desktop sessions.
60. Click the **End Lab** button on the *qwikLAB*™ lab details page.

61. When prompted for confirmation, click **OK**.
62. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS

Monitoring
Linux Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

In this lab you are going to develop a monitoring solution using CloudWatch and 3rd party services. You are going to learn how to interact with several AWS services using the CLI and the web console, and how to integrate with third party monitoring solutions.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with the Linux shell and using SSH to access Linux servers.

Topics Covered

This lab will demonstrate how to:

- Write and read monitoring statistics from CloudWatch
- Create alerts and notifications
- Integrate with 3rd party monitoring solutions (enterprise or cloud based)

Interacting with CloudWatch

Overview

Amazon CloudWatch provides monitoring for AWS cloud resources and the applications customers run on AWS. Developers and system administrators can use it to collect and track metrics, gain insight, and react immediately to keep their applications and businesses running smoothly. Amazon CloudWatch monitors AWS resources such as Amazon EC2 and Amazon RDS DB instances, and can also monitor custom metrics generated by a customer's applications and services. With Amazon CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.

In this section, you are going to interact with CloudWatch by writing and reading monitoring statistics.

Writing and Reading Monitoring Statistics

CloudWatch automatically collects statistics from your AWS resources, such as CPU utilization from EC2 instances, latency from ELBs, and other metrics from several services. Moreover, you can send your own statistics from your instance or application to CloudWatch, thus improving and customizing your monitoring solution.

The first section of this lab will send and retrieve a custom metric to CloudWatch.

Create an EC2 IAM Role

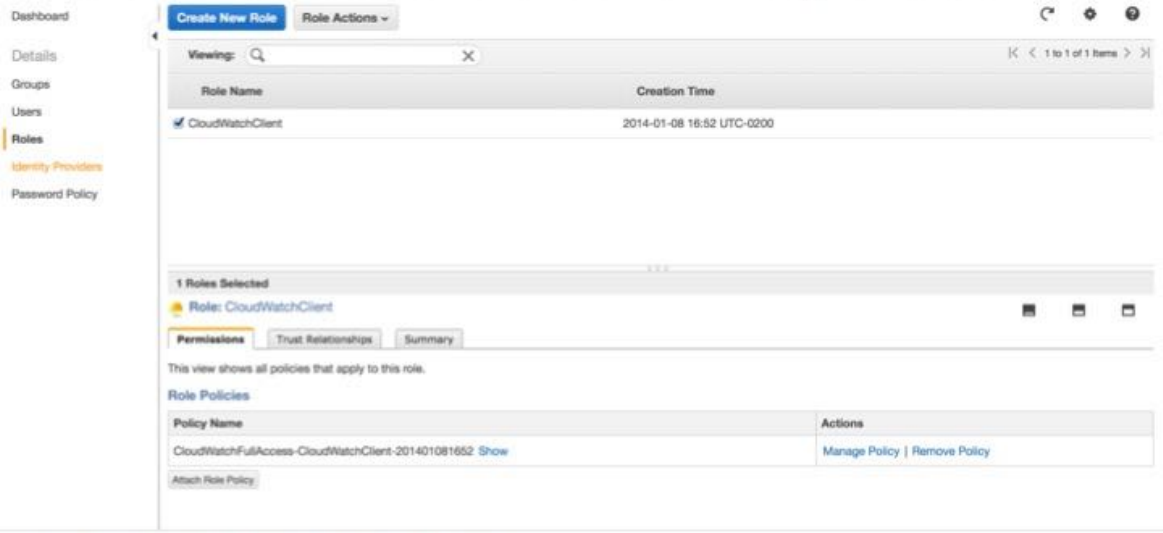
The monitoring client will need authorization to write statistics to CloudWatch. Using IAM Roles you can add the proper authorization policy without setting explicit credentials, providing convenience and security. To create an IAM Role with write access to CloudWatch, follow these steps:

1. Navigate to the IAM Management Console
2. Select **Roles** in the navigation menu
3. Start the **Create New Role** tool
4. Name your new role (e.g. CloudWatchClient)
5. In the **AWS Service Role** list, select **Amazon EC2**, indicating that this role will be used by EC2 instances

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

6. In the **Policy Template** list, select the **CloudWatch Full Access template,** granting read and write access to CloudWatch.
7. Review the policy document. It allows the API actions required to read and write to CloudWatch.
8. **Continue** and **Create Role**, finishing the role creation tool.

In the end of this step, you should see the new role in the IAM Management Console:

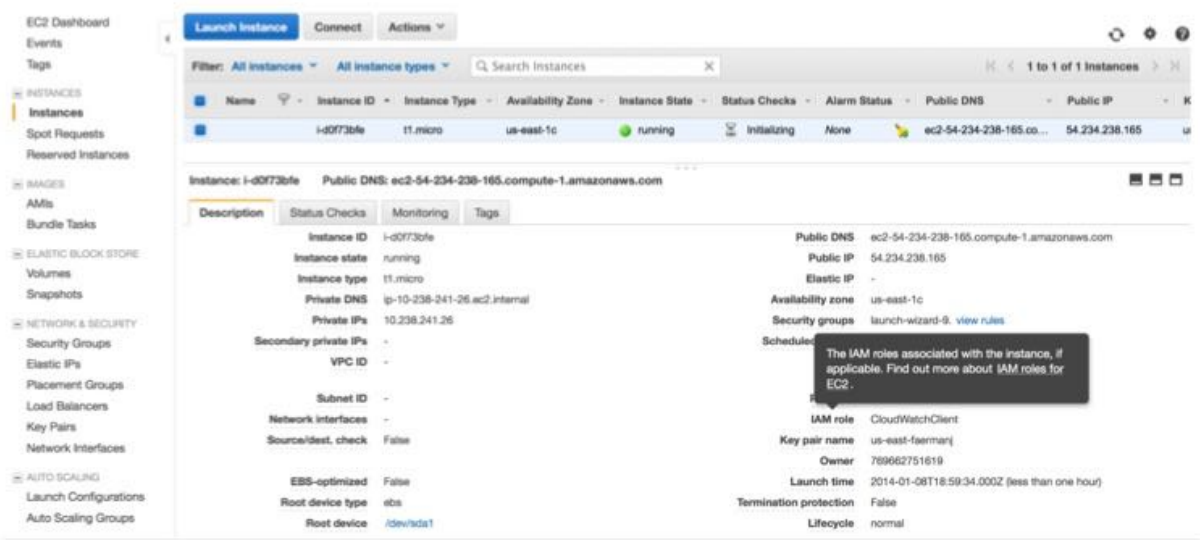


Launch an EC2 Amazon Linux Instance

9. Navigate to the EC2 Management Console
10. Click **Launch Instance**
11. Select the **Amazon Linux** AMI
12. Select the **t1.micro** instance type
13. Proceed with **next: Configure Instance Details**
14. Select the default VPC as **Network** and any **Subnet**
15. Select the **IAM Role** created in the previous section
16. **Review and Launch** the instance, after selecting your key pair

At the end of this section you should see a new instance with the selected role. Wait until the instance is ready to proceed.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



Write the new metric data

17. Access the instance launched in step 16 using SSH
18. Set the AWS Region for CLI commands

```
export AWS_DEFAULT_REGION=<your-aws-region-id>
```

19. Send a new metric data to CloudWatch, with any meaning. As an example, measure your attention level:

```
aws cloudwatch put-metric-data \

  --namespace Student \

  --metric-name AttentionLevel \

  --value 8 \

  --debug
```

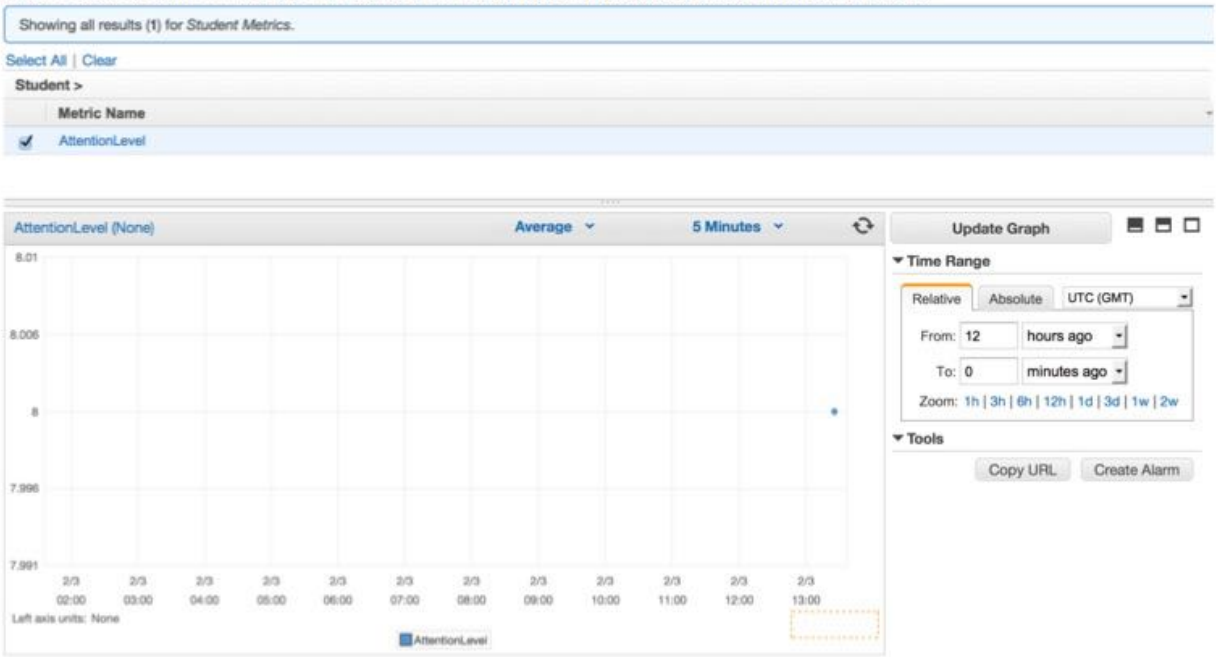
You should see a successful response (*HTTP Response Code: 200*) in the debug output.

Visualize the new metric

20. Open the CloudWatch Management Console
21. Select the *Student* namespace in the under **Custom Metrics** in the navigation menu, then open the *AttentionLevel* metric.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

In the end of this section you should see a graph displaying your custom metric.



Read statistics using the CLI

22. Get the minimum attention level statistic from the shell using the CLI:

```
aws cloudwatch get-metric-statistics \

 --metric-name "AttentionLevel" \

 --namespace="Student" \

 --start-time=$(date -d yesterday -I) \

 --end-time=$(date -d tomorrow -I) \

 --period=300 \

 --statistics="Minimum"
```

In the end of this section you should see your monitoring statistics in the console.

Alarms and Notifications

Create an alarm to notify you when the metric value drops below a given threshold:

23. Return to the browser widow displaying your custom metric graph
24. Start the **Create Alarm** tool
25. Configure your **Alarm Threshold**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Name: Attention too low
Description: Alert when attention level drops below 6
Whenever: AttentionLevel
is: < 6
for: 1 consecutive period

26. In the **Notification Action**

Whenever this alarm: Alarm is ALARM
Follow the create topic link
Send Notification To: SystemOperator
Email list: [your@email.com]

27. **Create Alarm**
28. Confirm the topic subscription request sent to your email
29. Generate a data point that triggers the alarm

```
aws cloudwatch put-metric-data \

  --namespace Student \

  --metric-name AttentionLevel \

  --value 4 \

  --debug
```

30. Navigate to your alarm and see that it has changed state from OK to ALARM
31. Wait for the email notification.

In the end of this section you should have received a low memory notification.

Integrating with Other Monitoring Services

Your monitoring solution may include 3rd party services, such as on-premise monitoring software or cloud monitoring services. In those cases, you have a couple options to register and unregister monitoring clients (your instances) as they are started and stopped.
To demonstrate the three scenarios presented in the course, you are going to start a "mock" monitoring server and then see how the different approaches work. The mock server is a simple registry of clients with an HTTP API, accepting requests with the following methods:

- GET: Lists the registered instance ids and registration time
- PUT or POST: Register the instance specified by the **instanceId** parameter
- DELETE: Clears the registry

Start the Monitoring Server

32. Open the instance terminal session (SSH)

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

33. Download and start the mock monitoring server

```
sudo -s

easy_install pip
```

```
pip install setuptools --no-use-wheel --upgrade

pip install Flask

mkdir /opt/awstc/

wget -O /opt/awstc/mon-server.py \

  https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-
monitoring-linux-v1.1/static/mon-server.py

python /opt/awstc/mon-server.py > mon-server.log 2>&1 &
```

34. Navigate to the Instances panel of the EC2 Management Console
35. Select your instance and, in the properties panel, copy the values for its **Security groups** and **Private IP** to a text editor
36. Navigate to the Security Groups panel of the EC2 Management Console
37. Select your instance's security group and copy its **Group ID** from the Details tab
38. In the **Inbound** tab, add a new rule allowing HTTP as protocol and paste the <Group ID> as source
39. **Apply Rule Changes**
40. Issue an HTTP GET request to your server using its Private IP (including the "http://"). In the next sections this value will be referenced as `<monitoring_server_url>`:

```
curl <monitoring_server_url>
```

In the end of this section you should see a message stating that no clients are registered.

Client Self-Registration

In this scenario, the monitoring clients will register themselves on their first boot:

41. Navigate to the Launch Configurations console
42. Start the **Create Auto Scaling group** tool
43. Proceed with the **Create Launch Configuration** tool

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

44. Select the **Amazon Linux AMI**
45. Select the **t1.micro** instance type
46. Proceed with **next: Configure Details**
47. Name your Launch Configuration (e.g. lcMonitoringLab)
48. Open the **Advanced Details** and paste the following **User data** script.

```
#!/bin/bash

curl -X PUT \

  -d instanceId=$(curl -s http://169.254.169.254/latest/meta-data/instance-
id) \

  <monitoring_server_url>
```

This command will register the instance with the mock server using a http PUT request with the instance id as a parameter. In a real scenario you would replace this with the client registration command from the actual monitoring service.

49. Replace `<monitoring_server_url>` with your mock monitoring server address (including http://)
50. Proceed with **next: Add Storage**, leaving default settings
51. Proceed with **next: Configure Security Group**
52. **Select an existing security group**, choosing the same security group as step 36
53. **Review** and **create Launch Configuration**, selecting your keypair
54. Conclude the **Create Lauch Configuration** tool. **Create Auto Scaling Group** tool will start
55. Name your Auto Scaling Group (e.g. asgMonitoringLab)
56. Set your **Group Size** to *start with 2 instances*
57. Select your default VPC as **Network**
58. Select any **Subnet**
59. Proceed to **Configure scaling policies**
60. Select **Keep this group at its initial size**
61. **Review**, **create your Auto Scaling Group** and **Close.**
62. Wait for the instances to be launched and ready.
63. In your SSH terminal session, list the registered instances with a new http GET request:

```
curl <monitoring_server_url>
```

At the end of this section you should see two instances registered at different times in the monitoring server.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Full Scan / Config Rewrite

In this scenario, the monitoring server (or any other instance) poll all instances and its configuration is rewritten periodically:

Grant EC2 access to your monitoring server

64. Navigate to the IAM Management Console
65. Select **Roles** in the navigation menu
66. Select the previously created role (e.g. MonitoringClient)
67. Open the **Attach Role Policy** tool
68. Select the **Amazon EC2 Read Only Access** policy template
69. **Apply policy**

Schedule the scan / config rewrite script

70. Access your monitoring server using SSH
71. Download the instance polling script and make it executable:

```
wget -O /opt/awstc/poll-instances.sh \

  https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-
monitoring-linux-v1.1/static/poll-instances.sh

chmod +x /opt/awstc/poll-instances.sh
```

72. Schedule the script to run every five minutes. Replace [monitoring_server_url] with the address of your monitoring server, including "http://".

```
echo "*/5 * * * * /opt/awstc/poll-instances.sh <monitoring_server_url>" |
crontab -
```

In the end of this section, you should see your instances updated in the monitoring server every five minutes with nearly the same registration time. You verify that with a HTTP GET request as in step 62.

Messaging (SNS and SQS)

In this scenario, you are going to push Auto Scaling events as messages to an SQS queue.

Create an SQS Queue for monitoring events

73. Navigate to the SQS Management Console
74. Start the **Create New Queue** tool
75. Name and create your queue (e.g. autoScalingEventsQueue) with default settings
76. Open the **Permissions** tab and **Add a Permission**
    Effect: Allow
    Principal: Check "Everybody (*)"
    Actions: Check "All SQS Actions (SQS:*)"
77. From the **Details** tab, copy to a text editor your **Queue ARN** and **URL**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Create an SNS Topic for monitoring events

78. Navigate to the SNS Management Console
79. Start the **Create New Topic** tool and name your topic (e.g. autoScalingEventsTopic)
80. **Create Subscription** for the new topic
    Protocol: Amazon SQS
    Endpoint: [Paste your **Queue ARN**]
81. Navigate to your Auto Scaling Groups
82. Select the auto scaling group created previously (asgMonitoringLab)
83. Open the **Notifications** tab
84. **Create Notification**
85. Select the auto scaling notifications topic (e.g. AutoScalingEventsTopic)
86. **Save** your notification

Grant SQS access to your monitoring client role

87. Navigate to the IAM Management Console
88. Select **Roles** in the navigation menu
89. Select the previously created role (e.g. MonitoringClient)
90. Open the **Attach Role Policy** tool
91. Select the **Amazon SQS Full Access** policy template

Consume Auto Scaling Event Messages

92. Access your terminal session on the monitoring server instance
93. Download the script that consumes event messages from SQS and make it executable:

```
wget -O /opt/awstc/consume-events.sh \

  https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-
monitoring-linux-v1.1/static/consume-events.sh

chmod +x /opt/awstc/consume-events.sh
```

94. Inspect and run the script, replacing the queue and monitoring server URLs

```
/opt/awstc/consume-events.sh <queue_url> <monitoring_server_url>
```

95. Navigate to your Auto Scaling Groups
96. Select the auto scaling group created previously (asgMonitoringLab)
97. **Edit** your auto scaling group from the **Actions** menu
98. Decrement the **Desired** and **Min** parameter to 1 and **Save**
99. Wait for the event messages to be consumed

In the end of this section you should see event messages in the console and the respective updates in the monitoring service.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Conclusion

Congratulations! You have successfully:

- Written and read custom monitoring statistics from CloudWatch
- Fired notifications from alarms
- Registered instances dynamically in 3rd party monitoring services

End Your Lab

100. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

101. Close any active SSH client sessions or remote desktop sessions.
102. Click the **End Lab** button on the *qwikLAB*™ lab details page.

103. When prompted for confirmation, click **OK**.
104. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operation on AWS
Backup and Archive
Linux Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

Traditional enterprise backup and recovery strategies typically take an agent-based approach whereby the entire contents of a server are backed up over either the local area network (LAN) or the storage area network (SAN). Traditional architectures have required this approach because replacing failed components is complex, time consuming, and operationally intensive. This has, in turn, created a backup environment that is complex to manage and resource intensive to operate—requiring technologies such as data de-duplication and virtual tape libraries to cope with ever-increasing workloads.

The AWS platform enables a far more lightweight approach to backup and recovery due, in part, to the following characteristics:

- Servers are now virtual abstract resources instantiated via code rather than being hardware-based.
- Capacity is available at incremental cost rather than up-front cost.
- Resource provisioning takes place in minutes, lending itself to real-time configuration.
- Server "images" are available on-demand, can be maintained by an organization, and can be activated

These characteristics offer you opportunities to recover deleted or corrupted data with less infrastructure overhead.

In this lab you will use the AWS Command Line Interface (CLI) tools to ship backups of a MySQL Database to an AWS S3 Bucket, retrieve one of the backups from S3 and perform a restore onto a MySQL Database. You will also use the AWS EBS Snapshot facility to take consistent backups of a striped database volume to S3 at a block level and restore the backup.

You will also see the advantage of using the Managed Backups feature of AWS RDS that automatically backs up your database, storing the backups for a user-defined retention period and enabling point-in-time recovery.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with the AWS Command Line Interface, basic Linux and MySQL commands. This lab is intended to follow modules that introduce Amazon EC2 and VPC.

Topics Covered

The following topics will be covered in the lab:

- Using RDS Managed Backups to Backup and Restore a MySQL RDS Instance
- Using an S3 bucket as a backup storage device for a MySQL Server Database
- Recovering a backup from S3 to MySQL Server using AWS CLI tools and restoring the Database
- Archiving the Backups to Glacier for Long Term Archival
- Using EBS consistent snapshots to backup a MySQL Database to S3

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

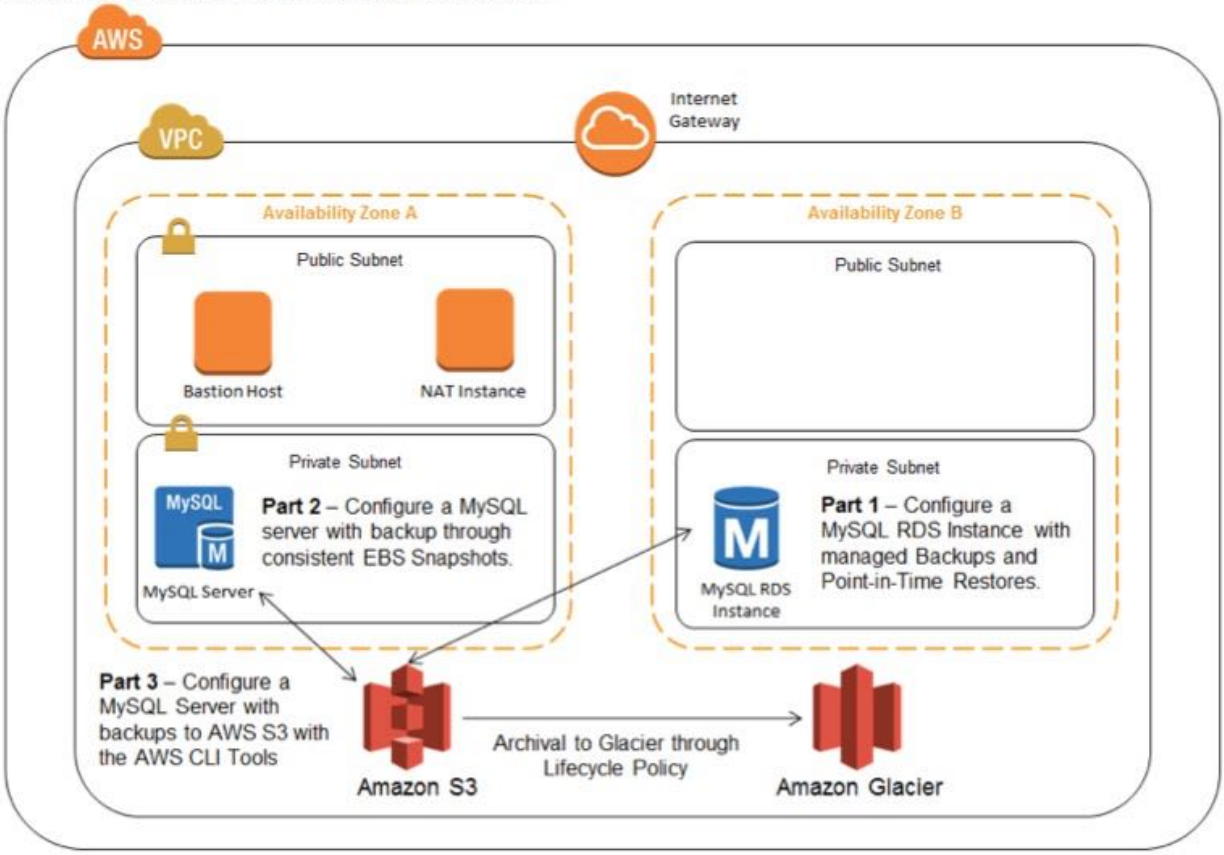
High Level Lab Flow

Given below is the high-level flow and architecture of the Lab. The VPC components of the Lab have been already covered in the earlier VPC lab and will be automatically created by this Lab template including a VPC, Private and Public Subnets, a Bastion Host and a NAT instance. The Lab template also creates an EC2 instance with MySQL Server installed and a MySQL RDS instance in the private subnet. There are three different scenarios that we will be covering on this Lab:

Part 1 – Point-In-Time Restore of an RDS Instance from an Automated Scheduled backup

Part 2 – Creating a scheduled MySQL Backups using mysqldump, Upload them to S3 for archival and DR purpose. Fetch a backup set from S3 and Restore the MySQL data to an earlier date

Part 3 – Creating a consistent Point-In-Time Snapshot of the MySQL Data, using it to restore Point-In-Time copy of the volume where the data resides



Note: This lab is not a comprehensive walkthrough of all possible backup and archival methodologies possible on AWS. The lab strives to give the user a basic understanding of common services that are used to achieve backups on AWS

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Part 1: Point-In-Time Restore Of an RDS Instance from An Automated Scheduled backup

AWS RDS (Relational Database Service) is a managed database service offering from AWS that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity. With RDS, AWS will manage time-consuming database administration tasks, freeing you up to focus on your applications and business. Backup is one of the database administration tasks that are taken care of by RDS. As part of the lab creation, an RDS instance has been created with a table that is being populated with a timestamp every second. We will see how to do a point-in-time restore of this RDS instance:

In this part you will:

- Connect to the Bastion Host and from there "Jump" to the MySQL EC2 Instance.
- Using the AWS CLI Restore An RDS Instance using the latest backup copy.

Connecting to the Bastion Host

The benefits of using a Bastion Host include:

- **Increased security**: by restricting access to your other resources to the Bastion Host only, there's less surface for attackers to hack into your infrastructure.
- **Auditing**: you could set up logging and auditing on the Bastion Host, so that every modification to your infrastructure would be securely logged and traceable. [*not covered on this lab*]
- **Easier to interact with the Services:** by using a Bastion Host for each infrastructure you manage on AWS, you don't need to rely on management software and settings being deployed on your own computer, on each of your company's computers, etc. All the tools are installed and configured in a central place – the Bastion Host – providing you with a homogeneous management environment that can easily be replicated.

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair
This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. To download the Private Key to your workstation navigate to the Lab Main Page and under "Connection Details" Click **download PEM/PPK** , For windows OS download the PPK version for MAC/Linux OS download the PEM version

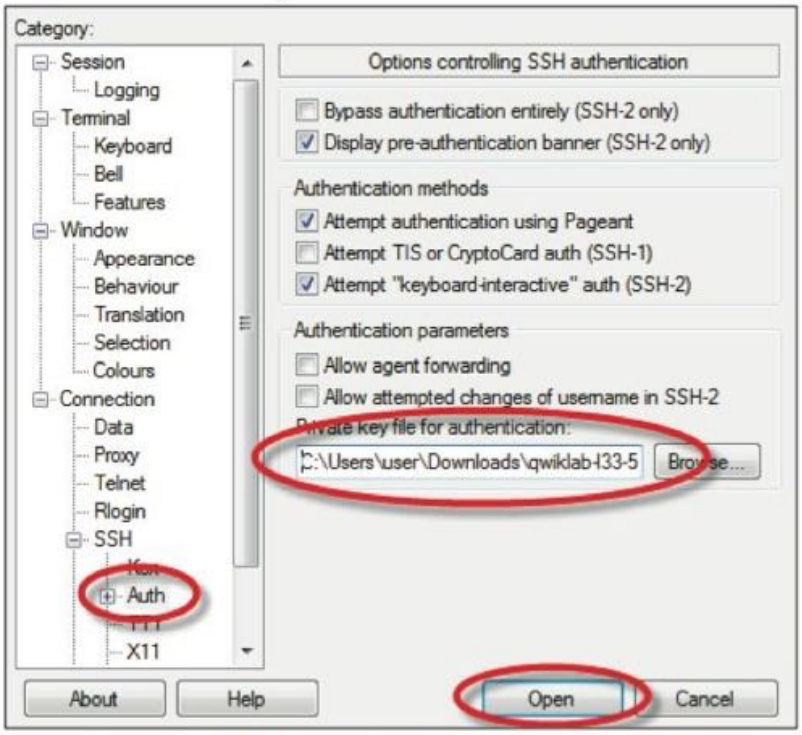
Obtain the public IP of the Bastion Host

1. Open the **AWS Management console**.
2. Select the **EC2** service.
3. From the menu on the left side, click **Instances**.
4. **Select the BastionHost instance** by clicking on the check box next to its name.
5. The Public IP is displayed on the bottom pane. Please write it down.

Connect to the Bastion Host
The steps to connect to the Bastion Host are different if you are using a Windows laptop or a Mac/Linux laptop. Please follow the appropriate set of instructions!
If you are using **Windows**:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

6. **Download PuTTY.exe** to a location of your choice (your Downloads folder, or your Desktop, for example) from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
7. **Double-click putty.exe** (the file you have just downloaded).
8. **Enter the Bastion Host's public IP into the Host Name input in Putty**. The public IP should have been obtained and written down in a previous section.
9. **Expand the SSH category** by double-clicking it. It is located on the menu on the left side.
10. Select the **Auth** sub-category (under the SSH category) by clicking it (not the + symbol next to it).
11. Click **Browser** and locate the PPK file (ending in **.ppk**). It is probably located in your Downloads folder.
12. Select the file and click **Open**.
13. In the **PuTTY** client, click **Open**.



14. When prompted, **click Yes** to allow PuTTY to cache the server's host key.

15. If you are prompted for a **username**, type **ec2-user**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

If you are using **Mac OS or Linux**:

16. Open the **Terminal** application.
17. Enter the commands below.
    In the **first command**, **change the path and file name** (~/Downloads/qwiklab-l33-5018.pem) to the location of the **.pem file** you downloaded earlier.
    In the **second command**, **change again the file name** and also **replace the string BASTION_IP** with the IP address of the Bastion Host you obtained earlier.

    chmod 400 ~/Downloads/qwiklab-l33-5018.pem
    ssh -i ~/Downloads/qwiklab-l33-5018.pem ec2-user@BASTION_IP

Connect to the MySQL EC2 Instance, by ssh from the bastion host a.k.a "Jumping"

Jump From The Bastion host

18. After logging in to the bastion host , Jump to the designated instance:

```
ssh -i ~/awsstudent.id.rsa ec2-user@My-SQL-EC2-Instance-Private-IP
```

Comment: Replace* My-SQL-EC2-Instance-Private-IP *with the target instance IP you wish to Jump into

But how do you get the IP Address of the MySQL EC2 Instance?

Hint: While Still Logged-In to the Bastion Host Using the AWS CLI, Describe the instances and apply a filter, based on the Instance tag value (The Lab created the Tag for you):

```
ec2-user$ aws ec2 describe-instances --filters 'Name=tag-
value,Values="MySQL Database Server"' --output table
```

19. From the output, record the IP address of the MySQL Instance by locating the **"PrivateIpAddress"** Key

As you could notice Tags are very powerful In assisting you to filter the API In order to extract information quickly , treat them as your campus!

20. Once you are logged in to the MySQL Instance, use the AWS CLI to describe the RDS Instance:

```
ec2-user$ aws rds describe-db-instances
```

Hint: The above command will return the output in JSON format, for your convenient You might opt for a table format Which is generally human friendly ☺, Just append "--output table" to the above command the result would be a key/value based table :

```
ec2-user$ aws rds describe-db-instances --output table
```

21. Examine the above output and the make sure that:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

- The DB instance is in an **"Available"** state by locating the key name **"DBInstanceStatus"**
- Record the **RDS DB Endpoint name**, The key name is "**Address**" Value is the **Endpoint** Name (We will need it later) , i.e. : "qahxj531yu5hv3.c0jhhreygsro.us-east-1.rds.amazonaws.com"
- Record the Value of the "**DBInstanceIdentifier**" key
- Record the Value of the **"DBSubnetGroupName"** key
- Records the Value of the **"VpcSecurityGroupId"** Key

Hint: Use the grep command to get the results faster, copy/paste the values to a text file!

22. While connected to the MySQL EC2 Instance, Type the below command to connect to the RDS instance. Replace **RDS-ENDPOINT** with the RDS endpoint **noted in Step 21**, Login to the MySQL Prompt (mysql>).

```
ec2-user$ mysql -u root -pAWSLabPassword$ -h <RDS-ENDPOINT>  AWSLab
```

23. Execute the SQL Statement below to confirm is the table has been created and data is being populated.

```
ec2-user$ Select * from table_timestamp;
```

Note the last date and time that the table was update!

Quit the MySQL prompt back to the Linux Shell (Just type "exit" and hit enter")

24. Next, let's attempt to recover the database to the latest point in time. When recovering from RDS Automated Back-up, **we actually create a new DB Instance using the last backup**. Use the below CLI command to restore the DB Instance to the latest point time:

```
ec2-user$ aws rds restore-db-instance-to-point-in-time --source-db-
instance-identifier <DBInstanceIdentifier-Recorded-On-Step3-Above> --
target-db-instance-identifier myRestoredDB --use-latest-restorable-
time --db-subnet-group-name <DBSubnetGroupName Recorded-On-Step21-
Above>
```

A successful init of a restore process should return a JSON output, The last key/value pair should be:

"DBInstanceIdentifier": "myrestoreddb"

---

25. The restore Process can take ~ 10-15 Minutes. **While this is in process <u>lets jump to Part 2</u>.** We will **resume Part 1 to validate our restore outcome at the end of Part 2).**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Part 2: Scheduled MySQL Backups using mysqldump and S3 As Backup Off-Site Storage

In this Part you will:

- List all S3 Buckets and Identify your off-site MySQL Target Backup Bucket
- Execute a bash script that will Configure, Create the MySQL Local Environment
- Use mysqldump to backup the MySQL database to a file, push the file to S3
- Restore the MySQL database from an earlier backup file
- Automate and Schedule the backup process

26. Login to the bastion host and from there Jump to the "MySQL EC2 Instance"

    Hint: You should already be familiar on how to get the IP Address of the MySQL Server, If you do not recall refer to Step 18

27. The Lab has created an S3 Bucket for you to store the off-site MySQL Backups we will need to record its name, from the MySQL EC2 Instance Linux shell list all buckets:

```
ec2-user$ aws s3 ls s3://
```

    Record the S3 Bucket name that starts with the following string: **"qlstack2-lab*"**

28. In this step we will execute a script that will configure and create our MySQL awslab database. On the Linux shell type the below commands (Must be executed under the ec2-user):

```
ec2-user$ cd ~
ec2-user$ sudo chmod +x ./sysops-backup-linux-mysql-db.sh
ec2-user$ sudo ./sysops-backup-linux-mysql-db.sh
```

Hint: The script will generate the below log file:

```
/home/ec2-user/backup-lab-mysql-db.log
```

If **errors** were reported from the script output, please advise your instructor & Refer to the log file

29. To Verify that the above script has created the database, tables and event examine the last output of the screen and identify a table similar to this:

```
+----------+---------------------+
| col_data | col_timestamp       |
+----------+---------------------+
| Data     | 2014-02-13 20:56:07 |
| Data     | 2014-02-13 20:56:08 |
| Data     | 2014-02-13 20:56:09 |
| Data     | 2014-02-13 20:56:10 |
| Data     | 2014-02-13 20:56:11 |
| Data     | 2014-02-13 20:56:12 |
+----------+---------------------+
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Which actions did the script performed?
The script created a MySQL Database **(Under /data)** named awslab , table named table_timestamp and Created an Event that will insert a row every second (**Do not be shy open the script and explore It!**)

30. Now let's backup our MySQL Database and Upload a copy of the backup to S3, We will use a script that will backup the awslab MySQL database to /backup , Type the below commands (Must be executed under the ec2-user):

```
ec2-user$ cd ~
ec2-user$ sudo chmod +x ./mysqlbackup.sh
ec2-user$ ./mysqlbackup.sh
```

If the script has executed successfully the output will be similar to this:

```
Backup of mysql db awslab db_backup20140214.133931.sql was completed
successfully! You can find me at /backup"
```

(**Do not be shy examine the script!**)

Now lets upload the backup files to Our S3 Bucket:

```
ec2-user$ cd /backup
ec2-user$ aws s3 cp /backup/ s3://<Insert-S3-Bucket-Name-From-Step-
27>/backups/db_backups/ --recursive
```

To verify that the backup files were successfully uploaded to the S3 Bucket, List all files in the bucket:

```
ec2-user$ aws s3 ls s3://<Insert-S3-Bucket-Name-From-Step-
27>/backups/db_backups/
```

The above command should list the S3 backup file !

Congratulations! You have created your first DB manual backup, but no one likes to run repetitive tasks manually ☹ it simply does not scale! , So how do we automate the Backup & S3 Upload Process?

31. To automate the process we will do the following:

-Execute the mysqlbackup.sh script again, but this time we will add the S3 Bucket Name as an **Argument** to The Script

```
ec2-user$ cd ~
ec2-user$ ./mysqlbackup.sh <Insert-S3-Bucket-Name-From-Step-27>
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

To verify that the backup and S3 Upload was successful, list the backup files on the bucket!(Did you forget how to list the files? Look at step 30 above).
If new files are listed It means that our process ran successfully, now we have a pretty good automated process we need to schedule it….

32. To schedule the backup every minute, execute the below:

Hint: Do not forget to replace **<Insert-S3-Bucket-Name-From-Step-27>**

```
ec2-user$ echo "* * * * *  /home/ec2-user/mysqlbackup.sh <Insert-S3-Bucket-Name-From-Step-27> 2>&1 >> /tmp/cronlog" > ec2cron
ec2-user$ sudo crontab -u ec2-user ec2cron
```

Verify the crontab you have just created:

```
ec2-user$ crontab -l
```

You Just created a fully automated MySQL S3 Off-site backup, Now just wait for a **few minutes** and list all the backup files on the S3 bucket , you will notice that new files are being added.

I just listed the content of the S3 Bucket but I cannot seem to locate new backup files, why?
To troubleshoot the automated process, examine the content of /tmp/cronlog and /tmp/awsclierr

33. Its time to verify our ability to **RESTORE** the MySQL Database to an earlier time!
We will:

- Create /backup/restore to store MySQL backup files
- List the files from the S3 Bucket
- Pick an old file and copy it locally
- Restore by creating a new Database

```
ec2-user$ mkdir -p /backup/restore
ec2-user$ aws s3 ls s3://<Insert-S3-Bucket-Name-From-Step-27>/backups/db_backups/
ec2-user$ aws s3 cp s3://<Insert-S3-Bucket-Name-From-Step-27>/backups/db_backups/<BACKUP-FILE> /backup/restore/res.sql
```

Remember: A Backup has no meaning if it is not restorable!, Always test your backups by Randomly Restoring them

34. Verify that the backup file has been retrieved to the restore folder by typing the command.

```
ec2-user$ ls -l /backup/restore
```

35. To perform the **RESTORE**, run the commands below. It will create a database called **awslabrestore** and restore the database.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

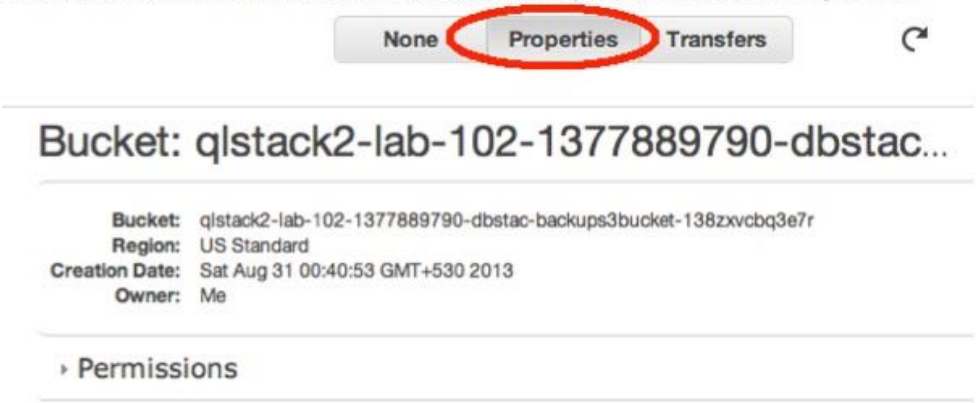
```
ec2-user$ mysql -u root -ppassw-lab -e "DROP DATABASE IF EXISTS
awslabrestore; "
ec2-user$ mysql -u root -ppassw-lab -e "CREATE DATABASE awslabrestore;
"
ec2-user$ mysql -u root -ppassw-lab awslabrestore <
/backup/restore/res.sql
```

36. Now its time to test our restore! :

```
ec2-user$ mysql -u root -ppassw-lab awslabrestore
When logged in to the mysql prompt execute a select:
mysql> select * from table timestamp;
```

Note the last line timestamp it should correspond to the backup file modified date we used to restore this Database

37. Exit from the SQL prompt by typing "**`exit;`**"
38. Next we will set a lifecycle policy for the backups so that you can archive older backups to a low cost archival storage called **AWS Glacier**. To do this go to the AWS Console and select the AWS S3 Service. Click the bucket that was noted in Step 27 and click on **Properties.**



39. Then click the "**Lifecycle**" link at the bottom of the page and Click the "**Add Rule**" button. Create a Rule like shown in the screenshot below to create a policy to archive all backups under the **"backups"** folder to Glacier after 30 Days.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

40. Congratulations, You have successfully setup backups to S3 for your MySQL Database and set a lifecycle policy for data to be archived to **AWS Glacier** after 30 days.

NOTE: In this example we are using a basic backup functionality that exports the entire database as a flat file SQL Dump. This is just for the sake of demonstration. In Production scenarios instead of SQL dumps, DBA's would be taking native full database and transaction log backups. These too can be similarly moved to S3 using the AWS command line scripts.

Part 1 (continued from step 25): Point-In-Time Restore Of an RDS Instance from An Automated Scheduled backup

Now it's time to check the status of the RDS Restore Instance

41. The point in time-restored DB instance will create the new DB with the "Default" security group. To make it accessible to other instances we need to modify the instance, We will need to:

    - Get the status of the restored DB making sure it is in an available state
    - Modify the Restored DB Instance Security Group to match the Source DB Instance's security group

    Get the status of the new Restored DB:

```
ec2-user$ aws rds describe-db-instances --db-instance-identifier
myrestoreddb --output table | grep DBInstanceStatus
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Get the source instance **security group ID** (Should start with sg):

```
ec2-user$ aws rds describe-db-instances --db-instance-identifier
<Recorded-On-Step21-Above> --output table| grep VpcSecurityGroupId
```

Change the restored Instance security group to match the source instance's SG:

```
ec2-user$ aws rds modify-db-instance --db-instance-identifier
myRestoredDB --vpc-security-group-ids <sg-XXXXX-Extracted-Above>
```

42. Record the **Endpoint** Name of the newly Restored RDS Instance:

```
ec2-user$ aws rds describe-db-instances --db-instance-identifier
myRestoredDB | grep Address
```

43. Next lets connect to the restored DB Instance, Type the below command to connect to the Restored RDS instance. Replace **RESTORED-RDS-ENDPOINT** with the RDS endpoint Recorded on step 42 above. You shall be logged in to the MySQL Prompt (mysql>).

```
ec2-user$ mysql -u root -pAWSLabPassword$ -h <RESTORED-RDS-ENDPOINT>
AWSLab
```

44. Execute the SQL Statement below to confirm is the data in the table has been recovered.

```
mysql> Select * from table_timestamp;
```

45. Congratulations you have used the backup and restore feature in RDS to recover a database to a point in time.
46. **Lets continue and jump to part 3**

Note: In the above lab, creation and configuration of the RDS Instance was done by the lab creation script for simplicity. Admins can configure and modify the backup retention and backup window of an RDS during creation.

Part 3: Restoring a MySQL Database Using a consistent point-in-time EBS Snapshot

Another method of performing backups on AWS is by using AWS Elastic Block Store (EBS) Snapshots. Snapshots are an excellent method for taking backups of entire servers. However for stateful machines like databases, where there is transactional data residing in memory, there are chances of data corruption if there is in-flight data that has not been committed to the disk. To perform consistent snapshots you need to pause writes to the database, flush the database writes to disk and then take a snapshot. The database that we created in the previous part on the MySQL instance resides on a single EBS Volume hence snapshots of the EBS volume should be taken while the database + file system are locked for writes. The following steps demonstrate how this can be done.

In this part you will:

- Lock the MySQL database, So only Read Operations will be allowed

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

- Flush and Lock the File system to block Write Operations
- Perform an EBS Snapshot of the MySQL EBS Volume
- Create a new EBS Volume from the snapshot taken above and mount it as the active MySQL Data Volume
- Start the Database and verify that the Restore was successful

47. Login to the MySQL EC2 Instance (Remember this should be done by "Jumping" from the Bastion Host).

48. Record the instance id of the Instance. This can be retrieved from within the instance using the instance metadata by typing the command:

```
ec2-user$ curl http://169.254.169.254/latest/meta-data/instance-id
```

49. The lab has created the EBS volume (/dev/sdf mounted as /data) for storing the MySQL data To operate on this EBS Volume we will need to extract Its Volume ID , As follows:

```
ec2-user$ aws ec2 describe-volumes --output text| grep <INSTANCE-ID>
```

Replace **<INSTANCE-ID>** with the instance id extracted on step 48 above.
The output of the above command should contains 3 Volumes, We are after "/dev/sdf", Record the Volume ID of "/dev/sdf" , this should be something like: **vol-xxxxxxxx**

50. Login to the MySQL Server with the command below.

```
ec2-user$ mysql -u root -ppassw-lab awslab
```

51. At the MySQL Prompt (**mysql>**) run the below commands.

```
FLUSH TABLES WITH READ LOCK;
SYSTEM sync
SYSTEM sudo xfs_freeze -f /data
```

52. Confirm if the writes have paused by running the sql statement **select * from table_timestamp;** You will see that the writes have paused. Note the timestamp of the last write for later reference. After this, Exit from the mysql prompt by typing the command "**exit;**" without the quotes.

53. Take a snapshot of volume: "/dev/sdf" , The command will require the EBS Volume ID Recorded in Step 49.

```
ec2-user$ aws ec2 create-snapshot --volume-id <vol-xxxxxxxx-Recorded-
Step-49> --description "Snapshot of /dev/sdf MySQL Database"
```

54. Verify that the snapshot is in a "completed" state:

```
ec2-user& aws ec2 describe-snapshots --output table --owner self
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

How do I recognize my snapshot?
Every snapshot has a key named: **VolumeId**, We already Recorded the volume ID on step 49. Once snapshot was located , **Record the Snapshot ID: snap-xxxxxxx**

Before Continuing to the next step verify that the snapshot is In **"Completed" state**!

55. Login to MySQL server using the command **"mysql -u root -ppassw-lab awslab"**. Resume writes to the database by executing the below commands.

```
SYSTEM sudo xfs_freeze -u /data
UNLOCK TABLES;
```

56. Verify that the writes have resumed by selecting data from the table **select * from table_timestamp;**
Exit from the **mysql>** prompt by typing "**exit;**"

57. Now assuming that the data has been corrupted, we will recover the database from the previously taken snapshots. To do this we need to detach the current volume by executing the below commands in the $ prompt. Replace **vol-xxxxxxxx** with actual volume id noted in **Step 49**.

```
ec2-user$ sudo /etc/init.d/mysqld stop
ec2-user$ sudo umount /dev/sdf
ec2-user$ aws ec2 detach-volume --volume-id vol-xxxxxxxx
```

58. Now lets create the New Volume from the snapshot. For that execute the below command in the $ prompt. Replace **<snap-xxxxxxx>** with corresponding snapshot ID's noted earlier (Step 54). Also replace **<AZ>** with the actual **availability zone in which the MySQL EC2 instance is running**

59. ***Hint: How do I get the Availability Zone of my Running MySQL EC2 Instance? Query the metadata to get this information:***

```
ec2-user$ curl http://169.254.169.254/latest/meta-
data/placement/availability-zone
```

Once the availability zone was recorded lets create the new EBS Volume from an existing snapshot:

```
ec2-user$ aws ec2 create-volume --snapshot-id <snap-xxxxxxx> --
availability-zone <AZ> --volume-type standard
```

From the above output **Record**:

- **VolumeId** (This is Our New Volume ID)
- **State** (Most likely at creation time the state will indicate creating/pending)

Before Continuing to the next step verify that the Volume is In **"Available" state**!

Hint: How do I verify that my newly create volume is in Available state?

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
ec2-user$ aws ec2 describe-volumes --volume-id <new-vol-xxxxxxx>
```

Replace the **<new-vol-xxxxxxx>** with the VolumeId Recorded at step 59 above.

60. Now let's attach the newly create volume to the Instance choosing the **device /dev/sdf**
    Replace **<INSTANCE>** with the Instance ID Recorded at step 48
    Replace the **<new-vol-xxxxxxx>** with the VolumeId Recorded at step 59 above.

```
ec2-user$ aws ec2 attach-volume --volume-id <new-vol-xxxxxxx> --
instance-id <INSTANCE> --device /dev/sdf
```

61. Mount the newly created volume and start the MySQL Server with the below commands.

```
ec2-user$ sudo mount -a
ec2-user$ sudo /etc/init.d/mysqld start
```

62. Login the MySQL server with the command.

```
ec2-user$ mysql -u root -ppassw-lab awslab
```

63. Select data from the table table_timestamp using the below SQL Statement and verify that the data is present till the time the snapshot was taken.

```
select * from table_timestamp;
```

Congratulations! You have successfully performed a consistent backup of your MySQL Data EBS Volume and were able to restore it!

Note: In this lab we have taken a manual snapshot of the data, but in actual production scenarios, the backup process would be automated using scripts and will be scheduled. It will also be implemented out-of-band on a different server than the server on which the MySQL Server is running.

End Your Lab

64. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

65. Close any active SSH client sessions or remote desktop sessions.
66. Click the **End Lab** button on the *qwikLAB*™ lab details page.

67. When prompted for confirmation, click **OK**.
68. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
Security - Event Collection
Linux Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

Security and network operations, incident management, audit, compliance, and forensics teams rely on logs to understand what is happening and what has happened across their infrastructure. The AWS infrastructure provides logging capabilities, which complement events from operating systems, applications and databases to give additional context when researching potential issues.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line tools.

Topics Covered

This lab will explore the methods used to obtain the logs from Amazon Relational Database Service (Amazon RDS). The methods used in this lab include:

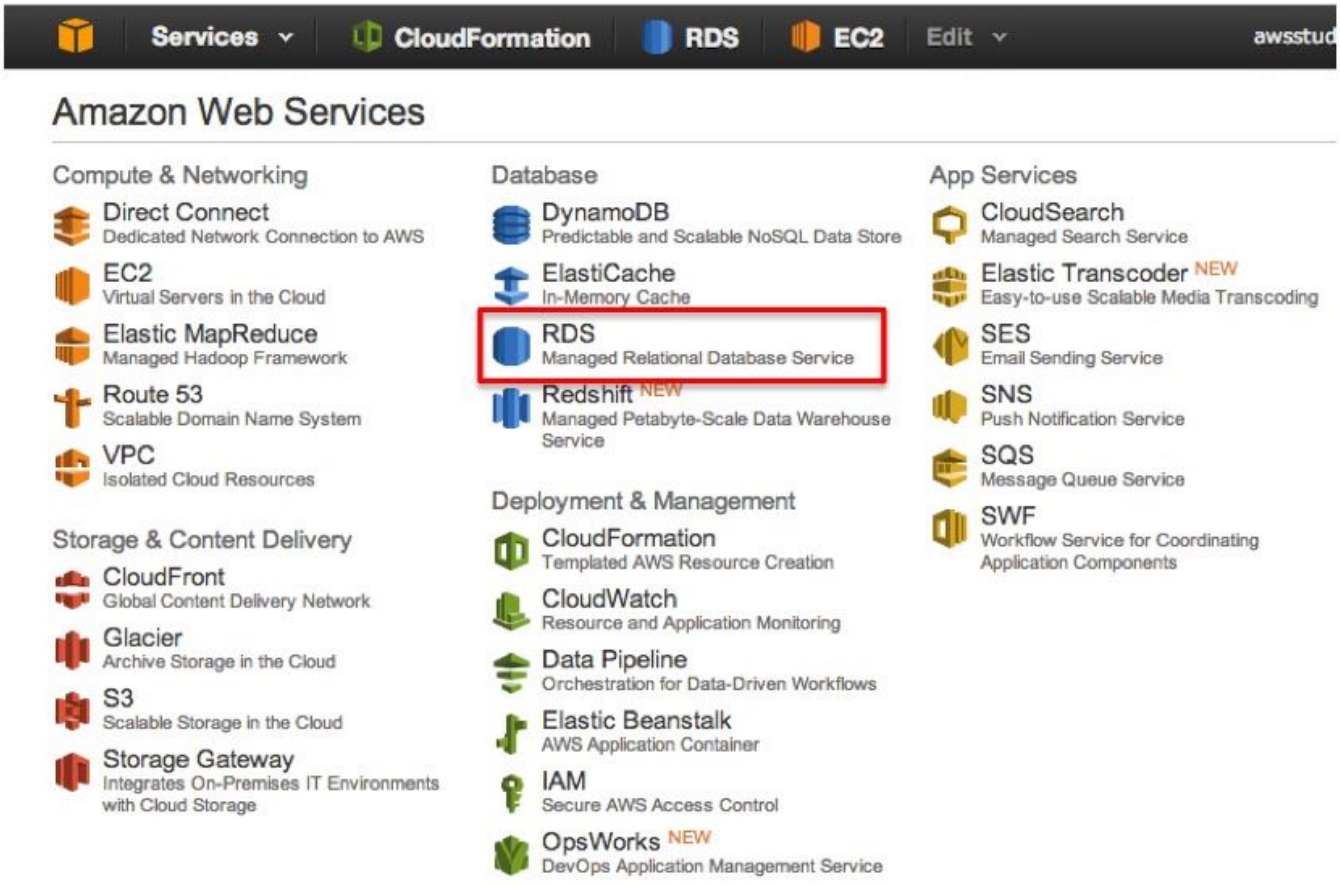
- Deploying an Amazon RDS instance and Amazon EC2 instance using CloudFormation
- Observing Amazon RDS events in the AWS Console
- Using the command line interface (CLI) to retrieve events from Amazon RDS

Viewing Events

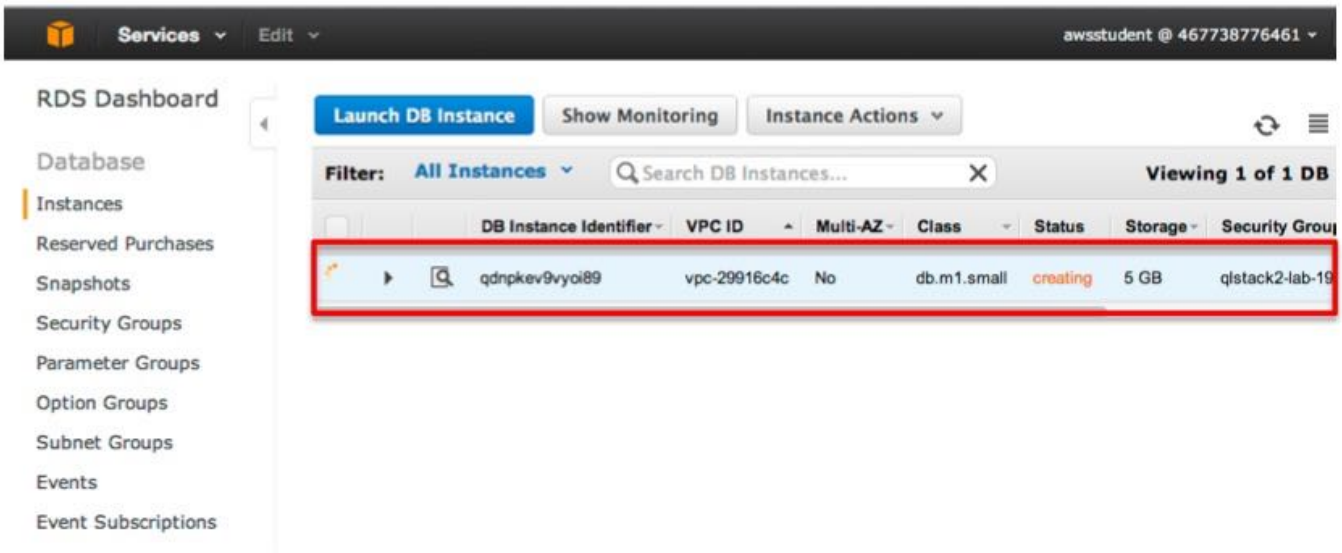
Locating Events with the AWS Console

1. In the AWS Management Console, select "RDS" as shown below.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

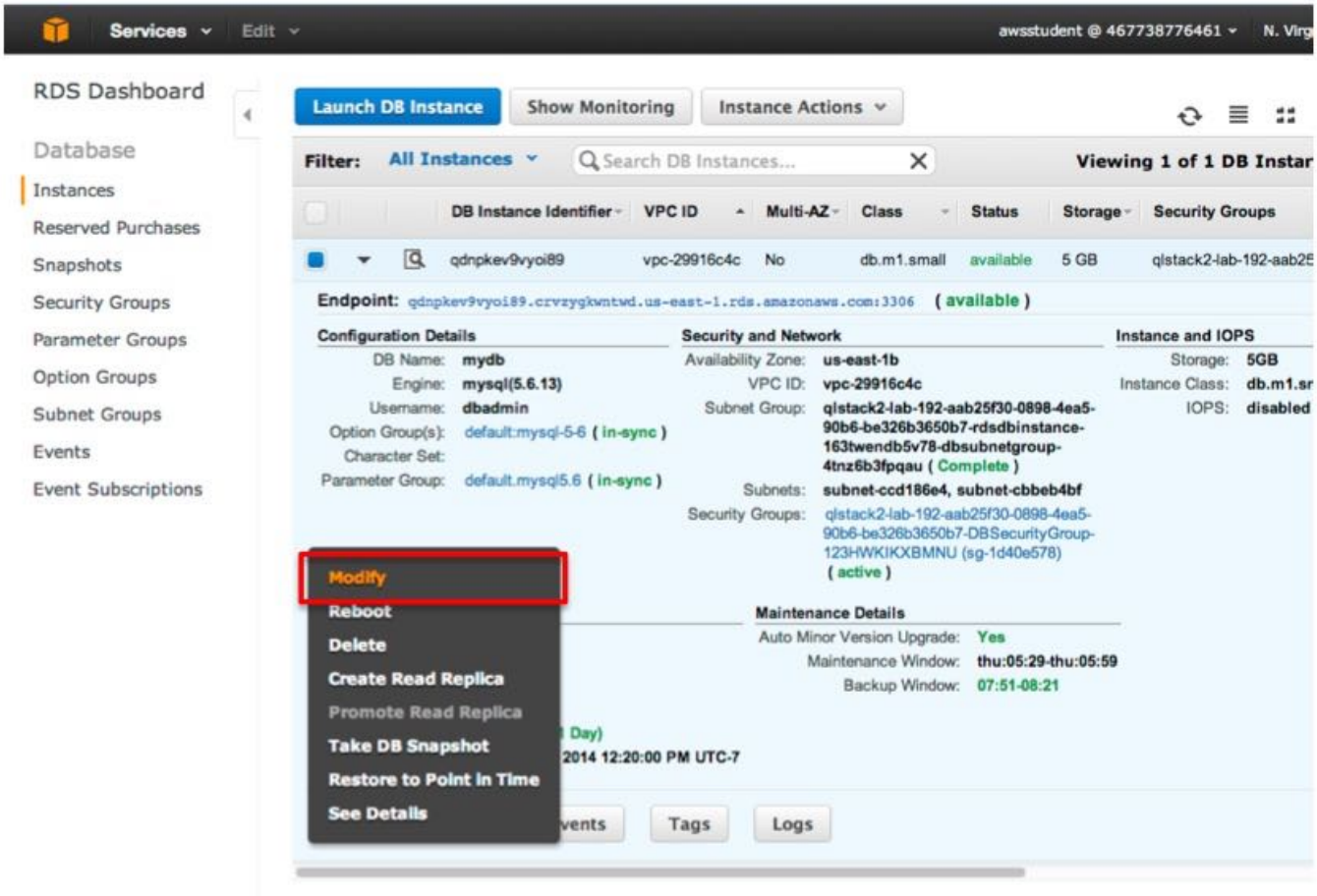


2. In the RDS Dashboard, you should see a new RDS database instance with a randomized identifier, similar to the screenshot below:



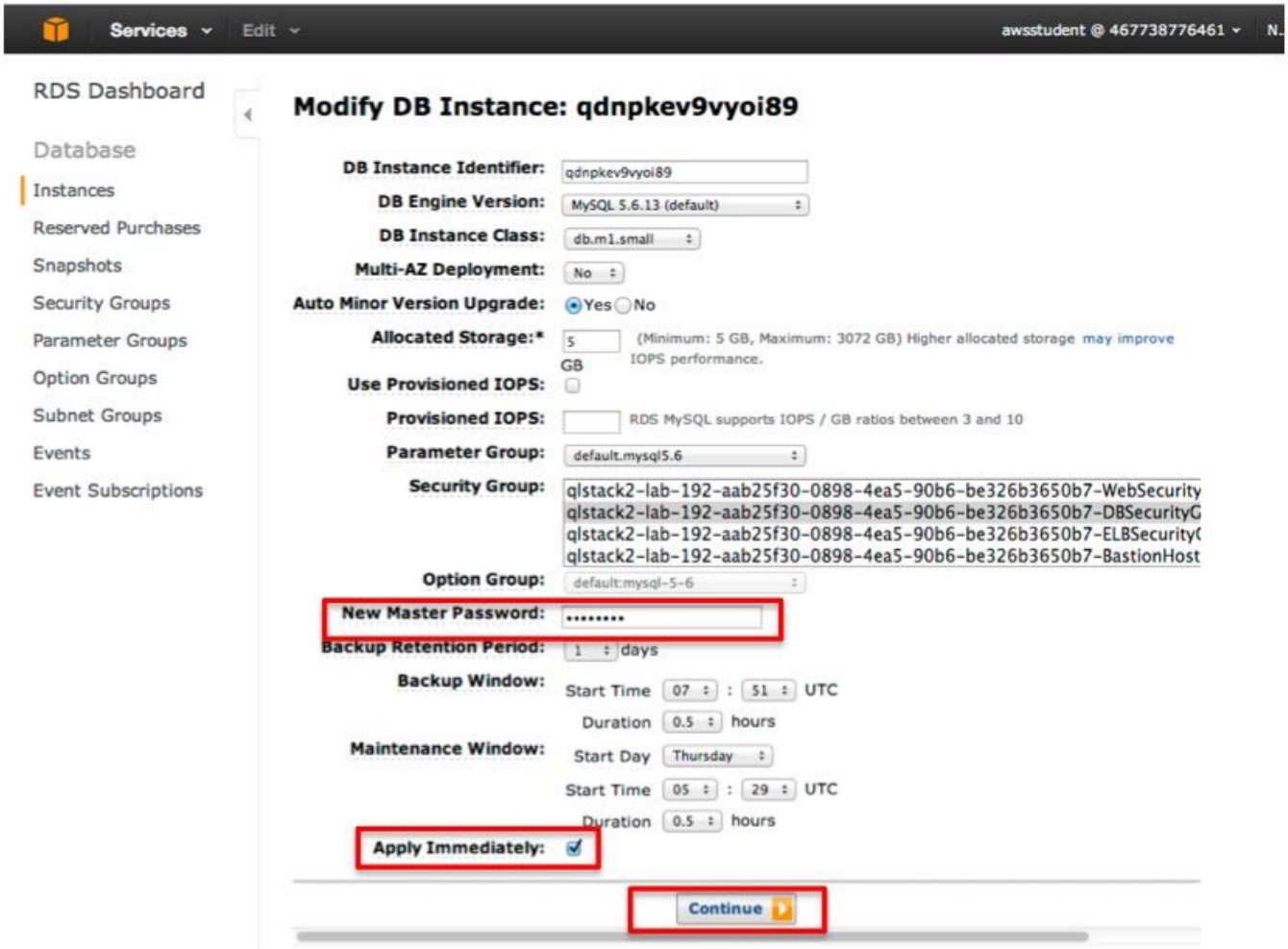
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

3. If the "Status" reads "Creating" (as shown above), wait until this process completes to proceed. After the database has been created, select the instance, and then choose "Modify" from the "Instance Actions" menu as shown below:



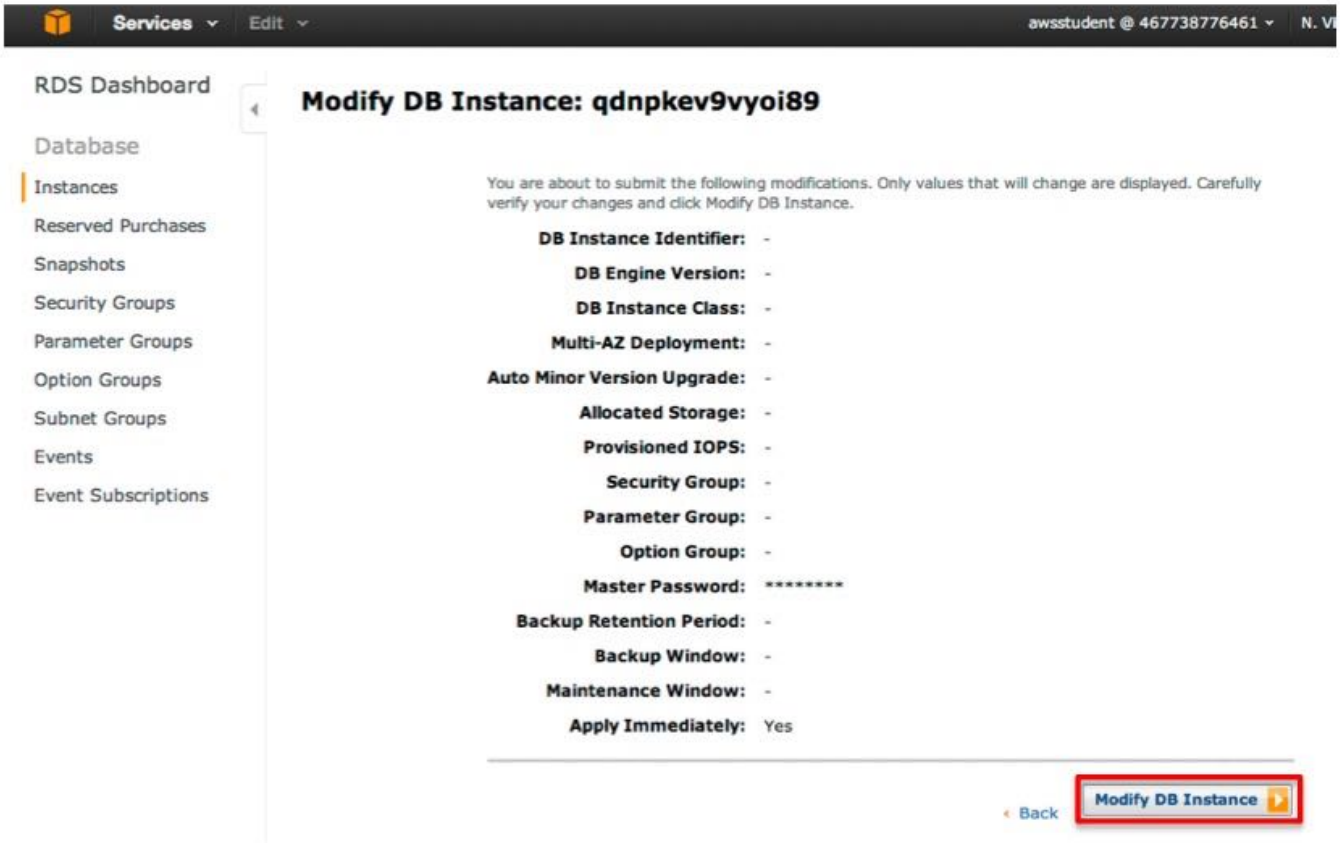
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

4. Now change the password for the database. Provide a new password in the "New Master Password" field, check the "Apply Immediately" box, and click Continue.



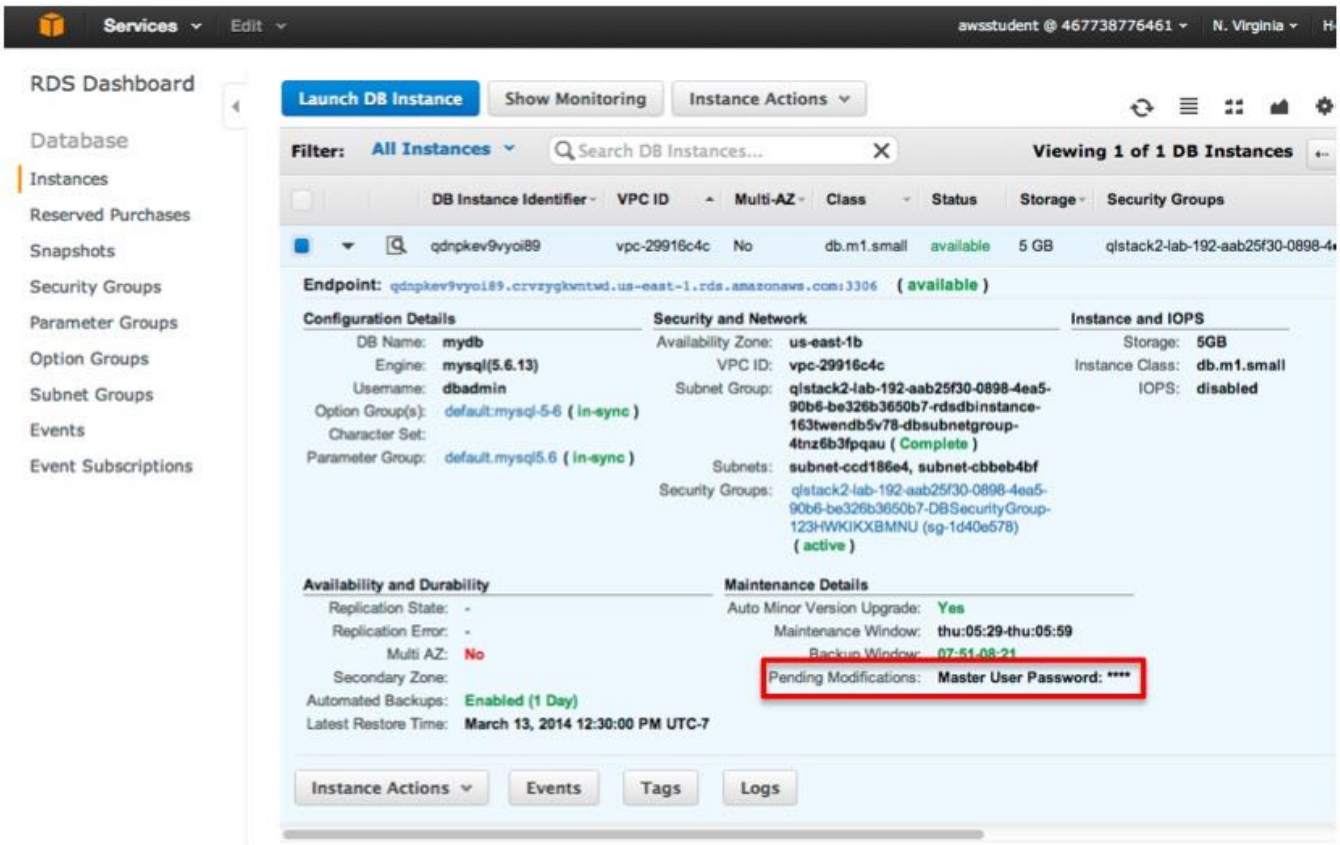
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

5. On the next screen, click "Modify DB Instance" to confirm these changes:



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

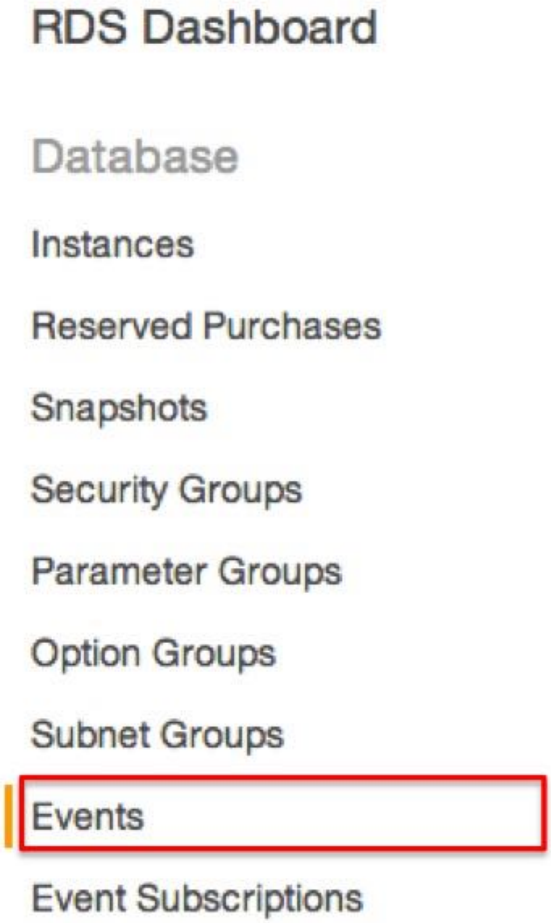
6. When you return to the instance details tab, you should see "Master User Password: ****" in the "Pending Notifications" field as shown below. Wait until this notification clears to proceed.



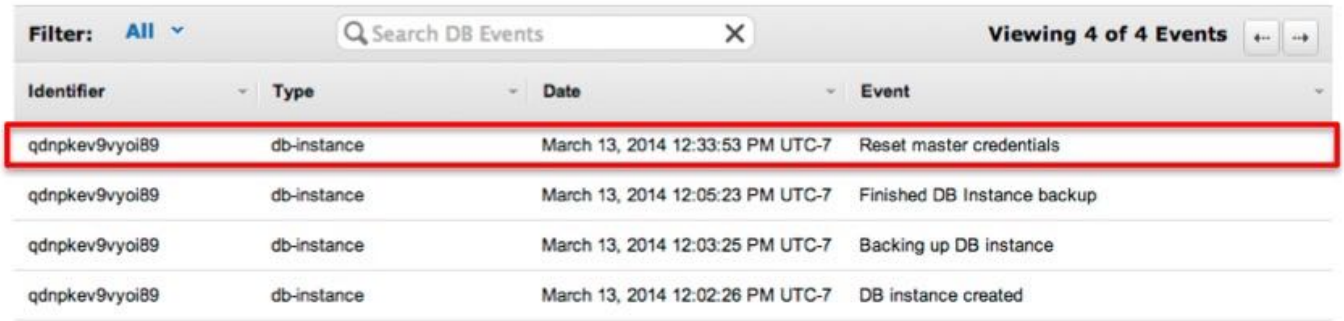
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

A password change to a database (especially a master password for the RDBMS engine) is typically deemed a significant security event in many organizations. An investigation could be warranted to determine if the change was authorized, and if the change was executed in coordination with change management approval. RDS has an event log, which can be used to provide information to such a query.

7. Click on the "Events" button in the left hand navigation menu of the RDS Dashboard as shown below.



8. Notice that some events were created when the RDS instance was deployed, including a modification the master credentials.



Identifier	Type	Date	Event
qdnpkev9vyoi89	db-instance	March 13, 2014 12:33:53 PM UTC-7	Reset master credentials
qdnpkev9vyoi89	db-instance	March 13, 2014 12:05:23 PM UTC-7	Finished DB Instance backup
qdnpkev9vyoi89	db-instance	March 13, 2014 12:03:25 PM UTC-7	Backing up DB instance
qdnpkev9vyoi89	db-instance	March 13, 2014 12:02:26 PM UTC-7	DB instance created

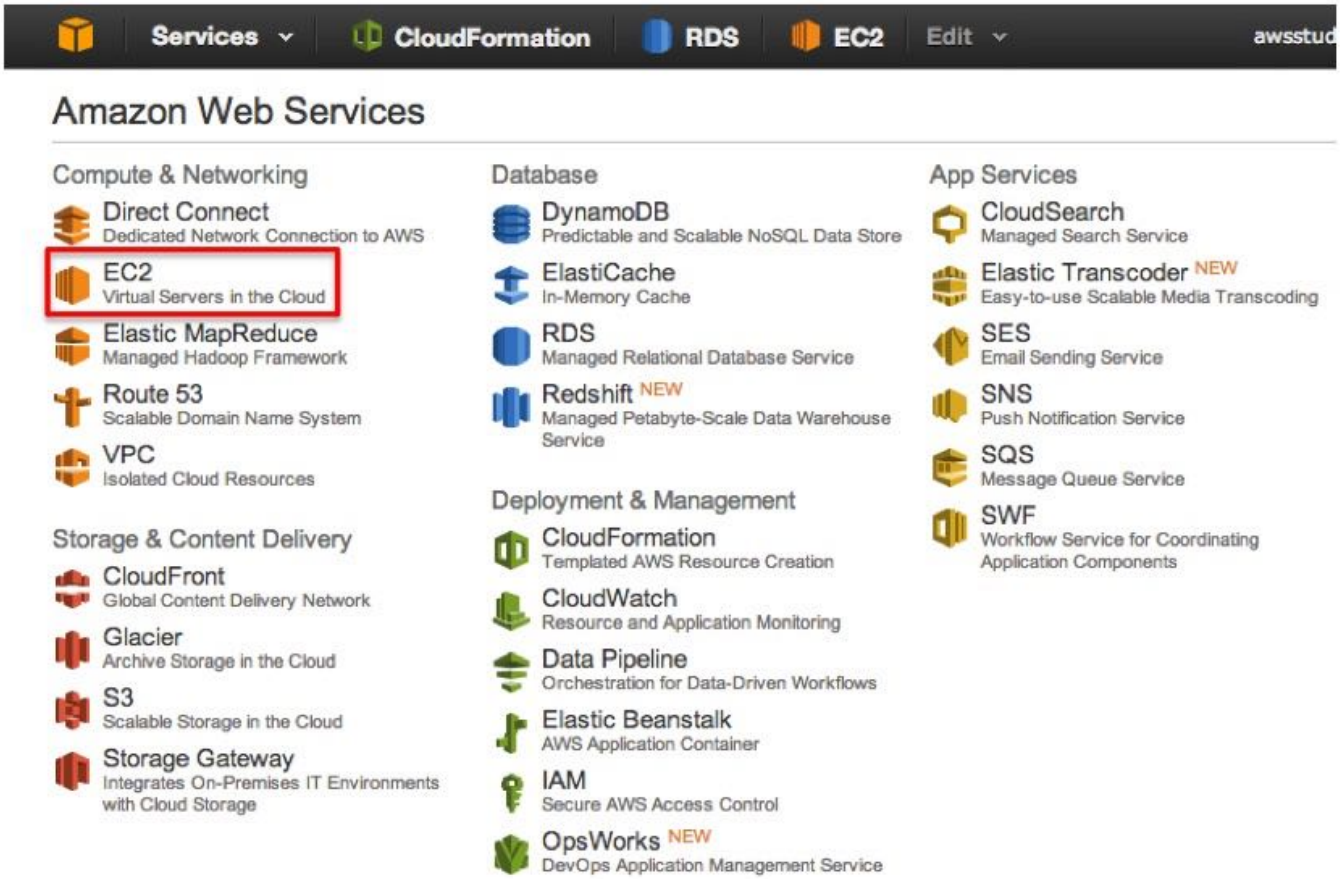
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Other services such as Amazon Redshift, CloudFormation, Elastic Beanstalk, and Data Pipeline have similar views of Events in the AWS Console.

Getting Events with the AWS CLI

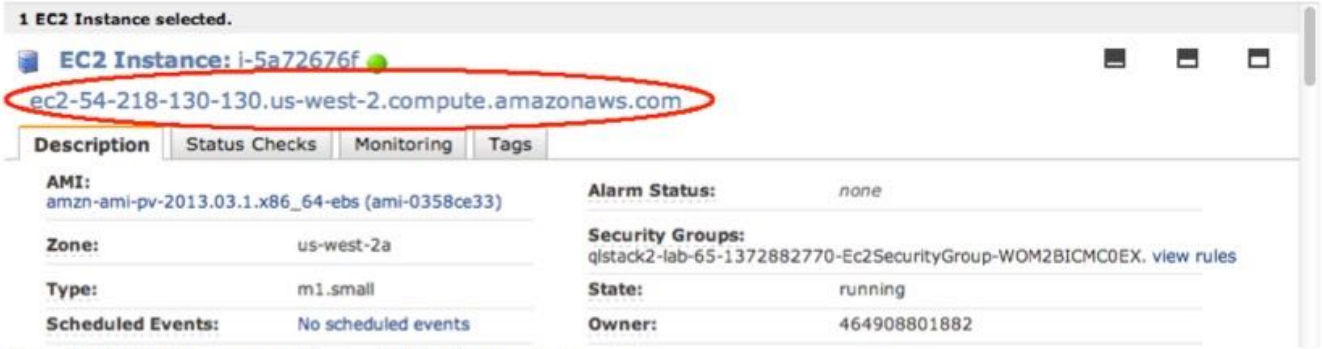
Events may be also retrieved using the AWS CLI. Programmatic access enables you to periodically collect these logs for analysis. For example, it is useful to send these events to a Security Information and Event Management (SIEM) platform for correlation and analysis by a Security Operations Center (SOC).

9. In the AWS Management Console, locate and connect to the EC2 instance as shown below.



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

10. Click on your instance and notice the fully qualified domain name (FQDN) in the detail window toward the middle of the screen.



Copy this FQDN, as we will need it for the next step.

Connect to your EC2 Instance via SSH (Windows Client)

11. See the appendix if you need assistance connecting to your instance via SSH.
12. Enter the following command to view RDS events:

```
aws rds describe-events
```

This should produce output similar to as shown below:

```
{
    "Events": [
        {
            "Date": "2014-03-13T19:02:26.210Z",
            "Message": "DB instance created",
            "SourceIdentifier": "qdnpkev9vyoi89",
            "EventCategories": [
                "creation"
            ],
            "SourceType": "db-instance"
        },
        {
            "Date": "2014-03-13T19:03:25.417Z",
            "Message": "Backing up DB instance",
            "SourceIdentifier": "qdnpkev9vyoi89",
            "EventCategories": [
                "backup"
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
            ],
            "SourceType": "db-instance"
        },
        {
            "Date": "2014-03-13T19:05:23.709Z",
            "Message": "Finished DB Instance backup",
            "SourceIdentifier": "qdnpkev9vyoi89",
            "EventCategories": [
                "backup"
            ],
            "SourceType": "db-instance"
        },
        {
            "Date": "2014-03-13T19:33:53.499Z",
            "Message": "Reset master credentials",
            "SourceIdentifier": "qdnpkev9vyoi89",
            "EventCategories": [
                "configuration change"
            ],
            "SourceType": "db-instance"
        }
    ]
}
```


Events can also be collected through the AWS API. For more information, visit http://aws.amazon.com/documentation/

Conclusion

Congratulations! You now have successfully:

- Viewed events for Amazon RDS in the AWS Console
- Viewed events for Amazon RDS using the AWS CLI

13. Close any active SSH client sessions or remote desktop sessions.
14. Click the **End Lab** button on the *qwikLAB*™ lab details page.

15. When prompted for confirmation, click **OK**.
16. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
Logging
Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

Although AWS EC2 instances are meant to be dynamic by design, the application or system logs on those resources may contain valuable information that should be preserved for potential later analysis. In this lab, we will walk through the process of creating and implementing an S3 bucket for aggregated log content.

In this lab you will:

- Create an S3 destination bucket for system logs
- Modify an IAM role to allow access to the S3 log destination
- Create a key schema for log prefixes
- Write and test a log rotation script to send logs to S3 from your centralized logging server

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with the Linux shell and using ssh to access Linux servers.

Topics Covered

This lab will focus on the creation of a script utilizing the AWS CLI. It will briefly touch on S3 bucket creation, S3 bucket lifecycle policies, and IAM Roles.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Connect to the Bastion Host

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair

This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this manual called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group

It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

1. Retrieve your current public IP address by pointing your browser to http://ip.appspot.com
2. Open the **AWS Management Console**.
3. Select the **VPC** service.
4. From the menu on the left side, click **Security Groups**.
5. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion host"**.
6. On the bottom pane, click the **Inbound** tab.
7. From the **Create a new rule dropdown**, select SSH.
8. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
9. Click the **Add rule** button.
10. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host

11. Open the **AWS Management console**.
12. Select the **EC2** service.
13. From the menu on the left side, click **Instances**.
14. **Select the BastionHost instance** by clicking on the check box next to its name.
15. The Public IP is displayed on the bottom pane. Please write it down.

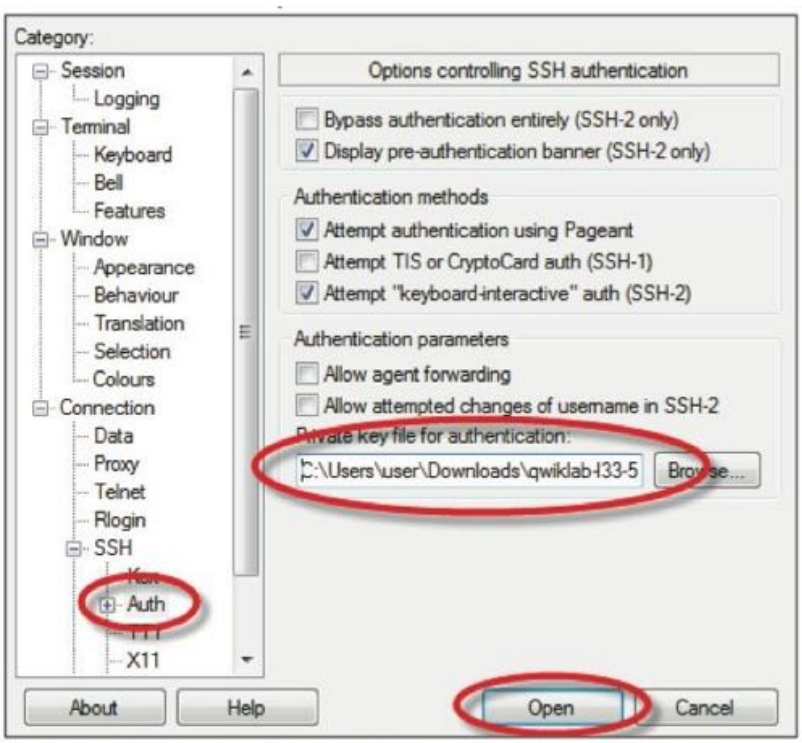
Connect to the Bastion Host

The steps to connect to the Bastion Host are different if you are using a Windows laptop or a Mac/Linux laptop. Please follow the appropriate set of instructions!
If you are using **Windows**:

16. **Download PuTTY.exe** to a location of your choice (your Downloads folder, or your Desktop, for example) from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
17. **Double-click putty.exe** (the file you have just downloaded).
18. **Enter the Bastion Host's public IP into the Host Name input in Putty**. The public IP should have been obtained and written down in a previous section.
19. **Expand the SSH category** by double-clicking it. It is located on the menu on the left side.
20. Select the **Auth** sub-category (under the SSH category) by clicking it (not the + symbol next to it).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

21. Click **Browser** and locate the PPK file (ending in **.ppk**). It is probably located in your Downloads folder.
22. Select the file and click **Open**.
23. In the **PuTTY** client, click **Open**.



24. When prompted, **click Yes** to allow PuTTY to cache the server's host key.

25. If you are prompted for a **username**, type **ec2-user**

If you are using **Mac OS or Linux**:

26. Open the **Terminal** application.
27. Enter the commands below.
    In the **first command**, **change the path and file name** (~/Downloads/qwiklab-l33-5018.pem) to the location of the **.pem file** you downloaded earlier.
    In the **second command**, **change again the file name** and also **replace the string BASTION_IP** with the IP address of the Bastion Host you obtained earlier.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
chmod 400 ~/Downloads/qwiklab-l33-5018.pem
ssh -i ~/Downloads/qwiklab-l33-5018.pem ec2-user@BASTION_IP
```

Create S3 log destination and allow access from the log server

Create S3 log destination bucket

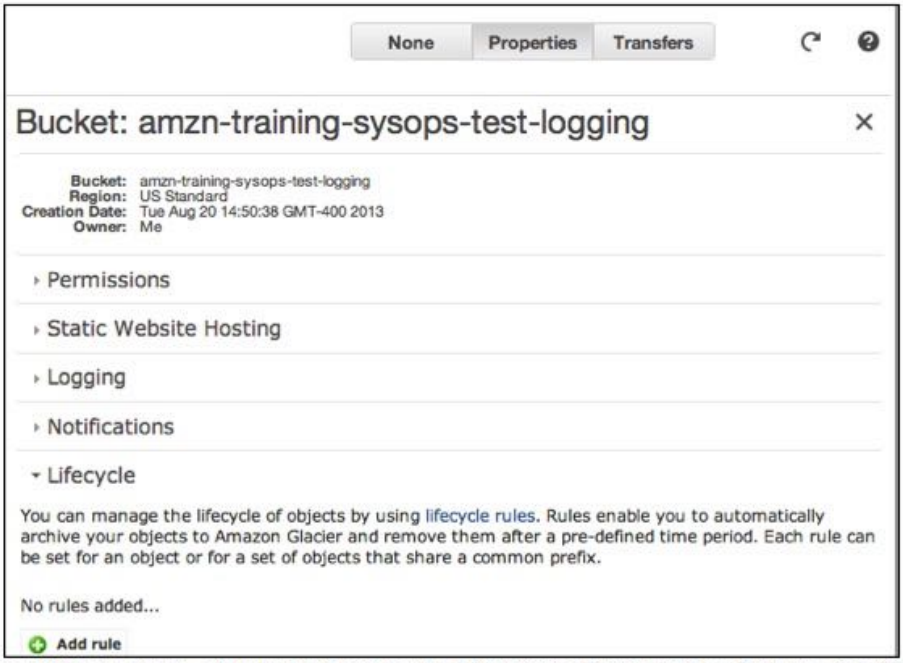
1. Log into the AWS Management Console and navigate to **S3.** Click on the "Create Bucket" button. Give your new bucket a meaningful name, something like "[yourcompanyname]-test-logging" - keep in mind that bucket names are globally unique, so you won't be able to create a bucket called "logging". Select the region that is closest to your primary EC2 infrastructure. If you plan to implement this in multiple regions, you might consider creating one bucket in each region.

Set bucket lifecycle policy

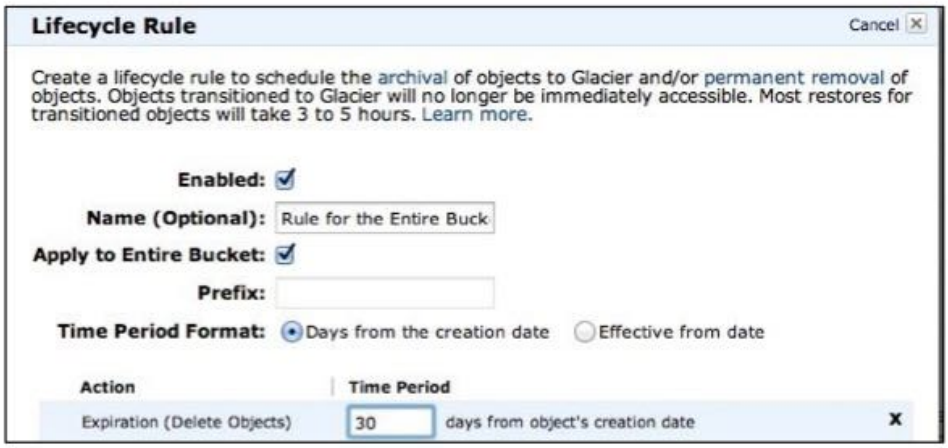
Since logs are generally only useful for a short time, this bucket should have a lifecycle policy to periodically delete older objects. In this lab, we will set the lifecycle of all log files at 30 days.

2. Select the new bucket in the S3 Console and click the "Properties" button at the top-right of the console. Click on "Lifecycle" and then click on the "Add rule" button.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



3. In the resulting screen, check the box marked "Apply to Entire Bucket", then click "Expiration". Enter "30" in the text box for "Time Period" and click the "Save" button. Click "OK" on the notification.



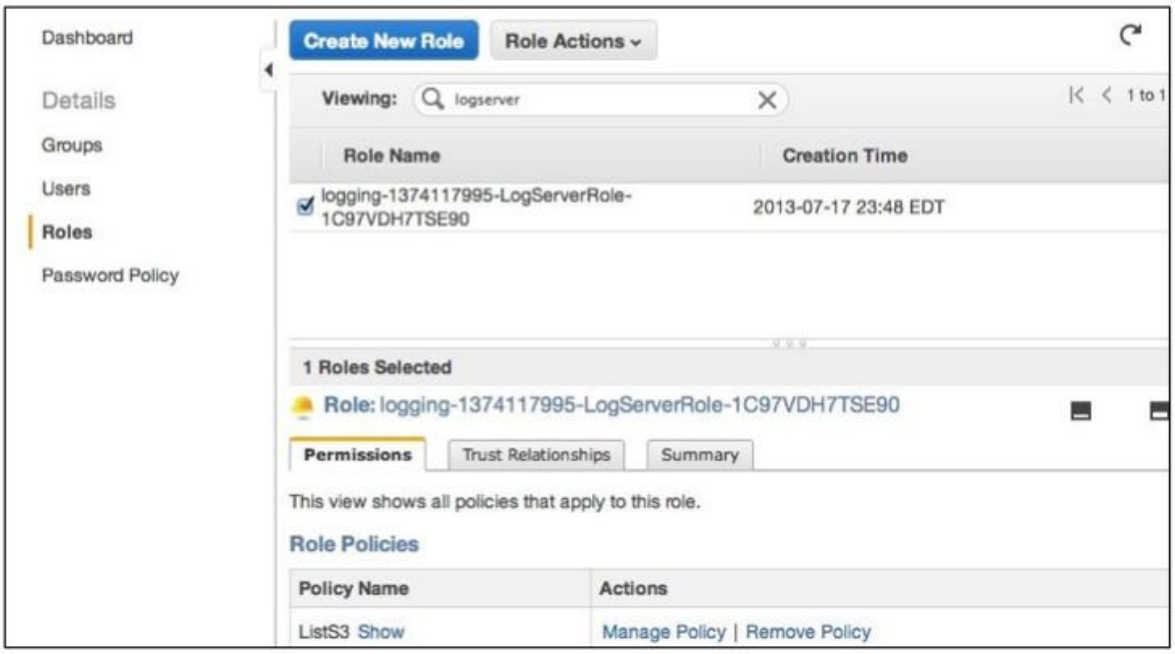
Create IAM Role Policy

The bucket is currently only accessible when accessing the S3 API or AWS Management Console with credentials. Since it needs to be accessible by the log server, we need to enable the log server's IAM Role to access the S3 bucket. This can be done in one of 2 ways: modify the bucket policy to allow access from the IAM Role, or modify the IAM Role's policy to be able to access the bucket. In this lab, we are going to modify the IAM Role's policy to be able to access the bucket.

4. In the AWS Management Console window and navigate to **IAM.** Select "Roles" from the menu on the left.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

5. Search for the string "logserver" – one role should match. Select that role and switch to the "Permissions" tab in the bottom half of the console.



6. Under Role Policies, there should only be one policy, called "root". Click on the button at the bottom labeled "Attach Role Policy".
7. Click the second radio button labeled ""Policy Generator", and click "Select".
8. Leave "Effect" as "Allow"
9. Select "Amazon S3" from the "AWS Service" dropdown
10. In the "Actions" dropdown, check the box for "All Actions"
11. In the "Amazon Resource Name" text box, enter: `arn:aws:s3:::<logging-bucket-name>/*`, `arn:aws:s3:::<logging-bucket-name>` replacing **`<logging-bucket-name>`** with the name of your S3 bucket.
12. Click "Add Statement" and click "Continue"
13. The resulting JSON policy will be displayed, along with a suggested name. The policy name can be changed without impacting any functionality.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

14. Click "Apply Policy"
15. After modifying this policy, the log server should now have access to the S3 bucket.

Log into logging server and test S3 access

16. In order to log into the logging server, you will first need to log into the bastion host (see **Error! Reference source not found.** if you need assistance). Get a list of s3 buckets by entering the following command:

```
aws s3 ls
```

You should see your logging bucket in the list.

17. To confirm that you have write access, create a small file in a temporary directory and try uploading it to the bucket:

```
echo "hello" > /tmp/bucketTest.txt

aws s3 cp /tmp/bucketTest.txt s3://<your-bucket-name>/
```

18. Replace **<your-bucket-name>** appropriately. Be sure to include the trailing slash after the bucket name.
19. Refresh your S3 console to confirm that the object was uploaded.

Create a staging folder for logs

The easiest way to set up a log archiving script is to put all older logs into a temporary folder, upload the contents of that folder to S3, and finally delete the local copy. Since logrotate (the Linux log rotation program) supports a "postrotate" script, a script can be created to be run after logs have been rotated.

20. Create a folder called "/var/log/old_syslogs" to be used as the temporary staging area:

```
sudo mkdir /var/log/old_syslogs
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Create log archive script

21. Paste the following script into the file `/tmp/archive-syslogs.sh`. Replace **<your-bucket-name>** with the name of your bucket.

```
#!/bin/bash
BUCKETNAME="<your-bucket-name>"
OLD_LOGDIR="/var/log/old_syslogs"
region=`(curl http://169.254.169.254/latest/meta-data/placement/availability-
zone/ 2>/dev/null \ && echo) | sed 's/[a-z]$//'`

sleep 5
cd $OLD_LOGDIR
for logFile in *
do
  test -s "$logFile" || echo >> "$logFile"
  aws s3 mv "$logFile" s3://$BUCKETNAME/ --region $region >/dev/null

done
```

22. Copy the script to `/usr/local/bin/` and set its permissions to be executable:

```
sudo cp /tmp/archive-syslogs.sh /usr/local/bin/

sudo chmod a+x /usr/local/bin/archive-syslogs.sh
```

23. Test this script by creating a temporary file in the staging directory and running the script

```
sudo touch /var/log/old_syslogs/backup_script_works.txt

sudo /usr/local/bin/archive-syslogs.sh
```

24. Refresh the S3 console to confirm that the new file was uploaded, and check the directory to confirm that it was subsequently deleted:

```
sudo ls /var/log/old_syslogs/
```

Modify logrotate to use archive script

25. Finally, modify the logrotate configuration for system logs to utilize the new script. Copy and paste the following contents into `/etc/logrotate.d/syslog` (replace the previous contents):

```
/var/log/cron
/var/log/maillog
/var/log/messages
/var/log/secure
/var/log/spooler
{
    sharedscripts
    dateext
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
    # Set the log format to have the date in it.
    # Although similar, it doesn't have the same functionality as the
"date" command
    dateformat -%Y-%m-%d-%s
    # send rotated logs to the old_syslogs folder for S3 backup
    olddir /var/log/old_syslogs
    # After rotating, we fire off the script from the lab
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2>
/dev/null
        /usr/local/bin/archive-syslogs.sh 2>>/var/log/syslog-archive-
log.txt || true
    endscript
}
```

Manually trigger a log rotation

26. Test that the addition to the logrotate configuration worked by forcing a logrotate run:

```
sudo logrotate -f /etc/logrotate.conf
```

After it completes, the system logs should have been moved to your S3 bucket.

Improve to support compression and key distribution

At this point, everything is functional – there are two improvements we can (and should) make.

27. Compress logfiles before uploading them to S3. We can use a faster compression than default for efficiency if desired.
28. Create a prefix with the application, class of logfile, and date stamp.
29. Both of these changes can be implemented in the `/usr/local/bin/archive-syslogs.sh` script as follows (again, replace **<your-bucket-name>** with your bucket name)

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

In this script we are creating a log rotation with added compression.

30. The bucket name BUCKETNAME, OLD_LOGDIR, and region are all variables that the script uses. Notice the region variable is getting its information from the instance meta-data.

31. The script will upload the file from S3 and removing it from the local drive.

Note that with the following `gzip -3 "$logFile"` you are adding compression. Aside from this, the rest of the script is the same as the previous script.

```
#!/bin/bash
BUCKETNAME="naton-test-linux-logging"
OLD_LOGDIR="/var/log/old_syslogs"
sleep 5
cd $OLD_LOGDIR
region=`(curl http://169.254.169.254/latest/meta-
data/placement/availability-zone/ 2>/dev/null \
&& echo) | sed 's/[a-z]$//'`
for logFile in *
do
    gzip -3 "$logFile"
    prefix="opscourse-class/syslogs/`date +%Y/%m/%d/%H/%M`"
  aws s3 mv "${logFile}.gz" s3://$BUCKETNAME/$prefix/ --region
$region >/dev/null
done
```

32. Force a log rotation one more time:

```
sudo logrotate -f /etc/logrotate.conf
```

33. Check your S3 bucket to confirm that the scripts are working as expected.

Epilogue and bonus

Congratulations, you have set up log rotation to S3!

Bonus activity (answer on next page): Determine how you could ensure that all logs go to S3 if an instance is removed from an Auto Scaling group. (Hint: Auto Scaling will issue a shutdown command on the instance when it is removed from the Auto Scaling group.)

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Bonus activity: The trick here is to force a log rotation before shutting down. The best way to do this is in the `init.d` scripts. A quick way to do it would be to just create a symbolic link to a log rotation script in the appropriate `rc.d` directories. In order to do this, you would first create a script to force a log rotation – let's call it `/home/ec2-user/forceLogRotation.sh`. This script forces log rotation when the "stop" parameter is passed as an argument on system shutdown and also initializes a lock file used by the rc.d subsystem when the "start" parameter is passed on system startup.

```
#!/bin/bash
prog="forceLogRotate"
lockfile=/var/lock/subsys/$prog
if [ "$1" == 'start' ]; then
  touch $lockfile
elif [ "$1" == 'stop' ]; then
  /usr/sbin/logrotate -f /etc/logrotate.conf
fi
```

After that's done, change the script to be executable:

```
chmod a+x /home/ec2-user/forceLogRotation.sh
```

Finally, create symbolic links to register `forceLogRotation.sh` for appropriate startup and shutdown runlevels:

```
sudo ln -s /home/ec2-user/forceLogRotation.sh
/etc/rc0.d/K60forceLogRotate
sudo ln -s /home/ec2-user/forceLogRotation.sh
/etc/rc6.d/K60forceLogRotate
sudo ln -s /home/ec2-user/forceLogRotation.sh
/etc/rc2.d/S60forceLogRotate
sudo ln -s /home/ec2-user/forceLogRotation.sh
/etc/rc3.d/S60forceLogRotate
sudo ln -s /home/ec2-user/forceLogRotation.sh
/etc/rc4.d/S60forceLogRotate
sudo ln -s /home/ec2-user/forceLogRotation.sh
/etc/rc5.d/S60forceLogRotate
```

Test it out by removing all logs from the S3 bucket and rebooting the instance twice (the first reboot initializes the lock required for kill script execution on subsequent reboots).

End Your Lab

34. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

35. Close any active SSH client sessions or remote desktop sessions.
36. Click the **End Lab** button on the *qwikLAB*™ lab details page.

37. When prompted for confirmation, click **OK**.
38. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
Creating an Elastic Infrastructure
Linux Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

This lab is intended to follow modules that introduce Amazon EC2 and VPC. It intends to build on that knowledge for preparing an elastic infrastructure.
Note: Replace `<place-holder>` with data from your lab environment.
Note: You are expected to see JSON output when you run the provided commands, as JSON is the default output. You may find that running these commands with the aid of jq on Linux improves legibility and navigation.

Note: Generated on the bastion host will be 4 files: aws_cli_config, awsstudent.id.rsa, UserData and lab-details.txt. You may have seen aws_cli_config and awsstudent_id.rsa before. The new files are:

- Lab-details.txt - Will contain output for creating your AMI and Auto Scaling configuration.
- UserData will contain the data needed to create your AMI.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line tools.

Topics Covered

The following topics are covered in these lab instructions:

- Amazon EC2 – Amazon Machine Image for deployment
  - Using EC2 Instance UserData
  - Using a script with EC2 Instance UserData
- Elastic Load Balancing
  - Creating an Elastic Load Balancer
  - Configuring a Health Check
- Amazon EC2 – Auto Scaling
  - Auto Scaling Launch Configurations
  - Auto Scaling Groups
  - Auto Scaling Policies
- Cloud Watch
  - Cloud Watch Alarms for Auto Scaling.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Connect to the Bastion Host

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair

This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this manual called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group

It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

1. Retrieve your current public IP address by pointing your browser to http://ip.appspot.com
2. Open the **AWS Management Console**.
3. Select the **VPC** service.
4. From the menu on the left side, click **Security Groups**.
5. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion host"**.
6. On the bottom pane, click the **Inbound** tab.
7. From the **Create a new rule dropdown**, select SSH.
8. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
9. Click the **Add rule** button.
10. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host

11. Open the **AWS Management console**.
12. Select the **EC2** service.
13. From the menu on the left side, click **Instances**.
14. **Select the BastionHost instance** by clicking on the check box next to its name.
15. The Public IP is displayed on the bottom pane. Please write it down.

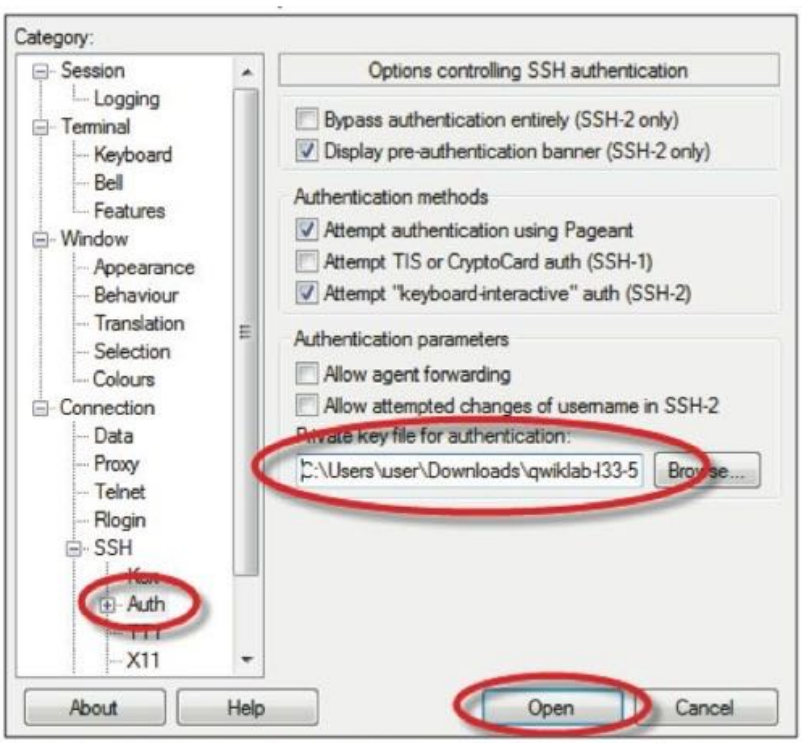
Connect to the Bastion Host

The steps to connect to the Bastion Host are different if you are using a Windows laptop or a Mac/Linux laptop. Please follow the appropriate set of instructions!
If you are using **Windows**:

16. **Download PuTTY.exe** to a location of your choice (your Downloads folder, or your Desktop, for example) from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
17. **Double-click putty.exe** (the file you have just downloaded).
18. **Enter the Bastion Host's public IP into the Host Name input in Putty**. The public IP should have been obtained and written down in a previous section.
19. **Expand the SSH category** by double-clicking it. It is located on the menu on the left side.
20. Select the **Auth** sub-category (under the SSH category) by clicking it (not the + symbol next to it).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

21. Click **Browser** and locate the PPK file (ending in **.ppk**). It is probably located in your Downloads folder.
22. Select the file and click **Open**.
23. In the **PuTTY** client, click **Open**.



24. When prompted, **click Yes** to allow PuTTY to cache the server's host key.

25. If you are prompted for a **username**, type **ec2-user**

If you are using **Mac OS or Linux**:

26. Open the **Terminal** application.
27. Enter the commands below.
    In the **first command**, **change the path and file name** (~/Downloads/qwiklab-l33-5018.pem) to the location of the **.pem file** you downloaded earlier.
    In the **second command**, **change again the file name** and also **replace the string BASTION_IP** with the IP address of the Bastion Host you obtained earlier.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
chmod 400 ~/Downloads/qwiklab-l33-5018.pem
ssh -i ~/Downloads/qwiklab-l33-5018.pem ec2-user@BASTION_IP
```

Amazon EC2 Instance and Custom AMI Creation

Deploying from Amazon AMI with Customizations

An essential piece of an elastic infrastructure is a base AMI, or image, which is to be used for auto-scaling configuration management. Some possible use cases for customer AMIs are: for keeping OS and application updates deployed consistently, keeping your code deployed consistently and as a means for replacing failed systems. An important aspect of this is that, should an update cause your application to fail, you would have the ability to easily roll back to a known good AMI. Another aspect, and the one we will be focusing on for this lab and the remainder of the day, is auto-scaling. Custom AMIs, along with idempotent customer application code allow for the auto-scaling functionality that Amazon AWS offers to function and provide you an elastic infrastructure.

Focus on building a custom AMI from an Amazon AMI. The example you will be working with is a Web application and will require some packages to be installed to support the deployment of the application, as well as some administrative tools. Included are OS updates, Apache, php, and other application specific items. The take away is that you can bake in all of your prerequisites into an image, as you might be doing now as a traditional operations engineer, and then deploy them in an automated fashion ensuring uniformity.

1. On the Bastion Host you will begin by creating an instance of an Amazon Linux AMI for a template of a web server instance. You will do this via the AWS CLI.
2. First let's look at the UserData file. On the bastion host execute the below command:

   ```
   base64 -d /home/ec2-user/UserData
   ```

   Take note of the last few lines of the file:

   ```
   # Cleanup Instance Secrets
   find -wholename /root/.*history -wholename /home/*/.*history -exec rm -
   f {} \;
   find / -name 'authorized_keys' -exec rm -f {} \;
   rm -rf /var/lib/cloud/data/scripts/*
   ```

   What is the significance of these lines? These lines clean out all of the history, authorized keys, and scripts which contain user data. This is done so that as instances are deployed from this AMI, they do not persist sensitive information. This is one way that is advisable to ensure that old traces of user information are cleaned up on an instance before AMI creation.
3. Retrieve AMI image-id and PublicSubnet information from /home/ec2-user/lab-details.txt

   ```
   cat /home/ec2-user/lab-details.txt
   ```

4. Create a Security Group with the **Name: ALL** and **Description: SSH, HTTP and HTTPS open**. Provide ingress access to ports SSH (22), HTTP (80) and HTTPS (443) from all addresses. The same can be done under **VPC > Security Groups** for your VPC. Provide access to all networks

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

(0.0.0.0/0). Do not forget to click on **Apply Rule Changes** for the changes to take effect. Make a note of the security group id available under **Details > Group ID** as you will use the same in the next step. One can similarly control outbound (egress) traffic if required. This option however is not available in EC2 classic mode.

5. Replace retrieved data from lab-details.txt with the <> sections. You could use any public subnet from the previous step. For the key pair do not use any extension.

```
aws ec2 run-instances --key-name <qwikLABS-xxxx-xxx> --
instance-type m1.small --image-id <AMIId> --security-group-
ids <ALLSecurityGroupID> --subnet-id <PublicSubnetID> --
user-data file:///home/ec2-user/UserData --associate-
public-ip-address
```

The resulting output will consist of useful data such as instance ID and keyname. Your output will look similar to the following:

```
...

"Instances": [
...
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2014-01-18T20:22:12.000Z",
            "PrivateIpAddress": "10.50.2.95",
            "ProductCodes": [],
            "VpcId": "vpc-a311f5c6",
            "StateTransitionReason": null,
            "InstanceId": "i-e1ec54c0",
            "ImageId": "ami-05355a6c",
            "PrivateDnsName": "ip-10-50-2-95.ec2.internal",
            "KeyName": "qlstack2-lab-135-1390075369",
            "SecurityGroups": [
                {
                    "GroupName": "qlstack2-lab-135-1390075369-
WebSecurityGroup-1UD5166X1HSYY",
                    "GroupId": "sg-5aa3663f"
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
                }
            ],
            "ClientToken": null,
            "SubnetId": "subnet-92b8e2d4",
            "InstanceType": "m1.small",
...
                "AvailabilityZone": "us-east-1a"
            },
...
```

6. On the management console find your way to **EC2 > Instances** and name the newly created instance as **AMI Instance** by clicking the **Name** column on the instance.

Converting Instances into Custom AMIs.

Having deployed an instance from the Amazon AMI with your customizations in the UserData file, you will now create a custom AMI to use for deployment of instances with Auto Scaling.

7. Gather your Instance ID from the output of the previous command or on the console: **EC2 > Instances** and note the Instance ID from the center top panel.
8. Check the instance creation state by replacing retrieved data from the previous output with the <> section.

```
aws ec2 describe-instance-status --instance-id <Instance-Id>
```

Note: You will not see any indication that bootstrapping is complete. The 'Running' state is equivalent to an 'instance is powered on' indicator. You could, at this point, your instance browser to the instance IP address and see a web page appear. Since you cleaned up the keys as part of bootstrapping, ssh access into the instance would be disabled. Be patient as this step could take a while. Proceed to the next step only after bootstrapping has completed successfully.

9. Stop the instance via the management console or using the following AWS CLI:

```
aws ec2 stop-instances --instance-ids <Instance-Id>
```

To which your output will look similar to the following:

```
{
"StoppingInstances": [
{

"InstanceId": "i-f9914680",
"CurrentState": {
   "Code": 64,
      "Name": "stopping"
      },
   "PreviousState": {
      "Code": 16,
      "Name": "running"
...
```

10. Create an AMI from the previously created instance. Replace retrieved data with the <> section.

```
aws ec2 create-image --name WEBSERVER --instance-id <Instance-Id>
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Broken down, this command calls `aws ec2` service. It then calls the `create-image` operation. And finally, it calls the parameters `instance-id` and `name`, which tags the image with a name. Your output will be similar to the following, indicating a successful custom AMI creation. Make a note of this ID as you will use the same in future.

```
"ImageId": "ami-a97c37c0"
```

11. Validate your AMI creation in the management console: **EC2 > Images > AMIs**.
    You can proceed to the next step while the AMI is getting created.

Amazon EC2 Auto Scaling

Understanding and Employing Auto Scaling

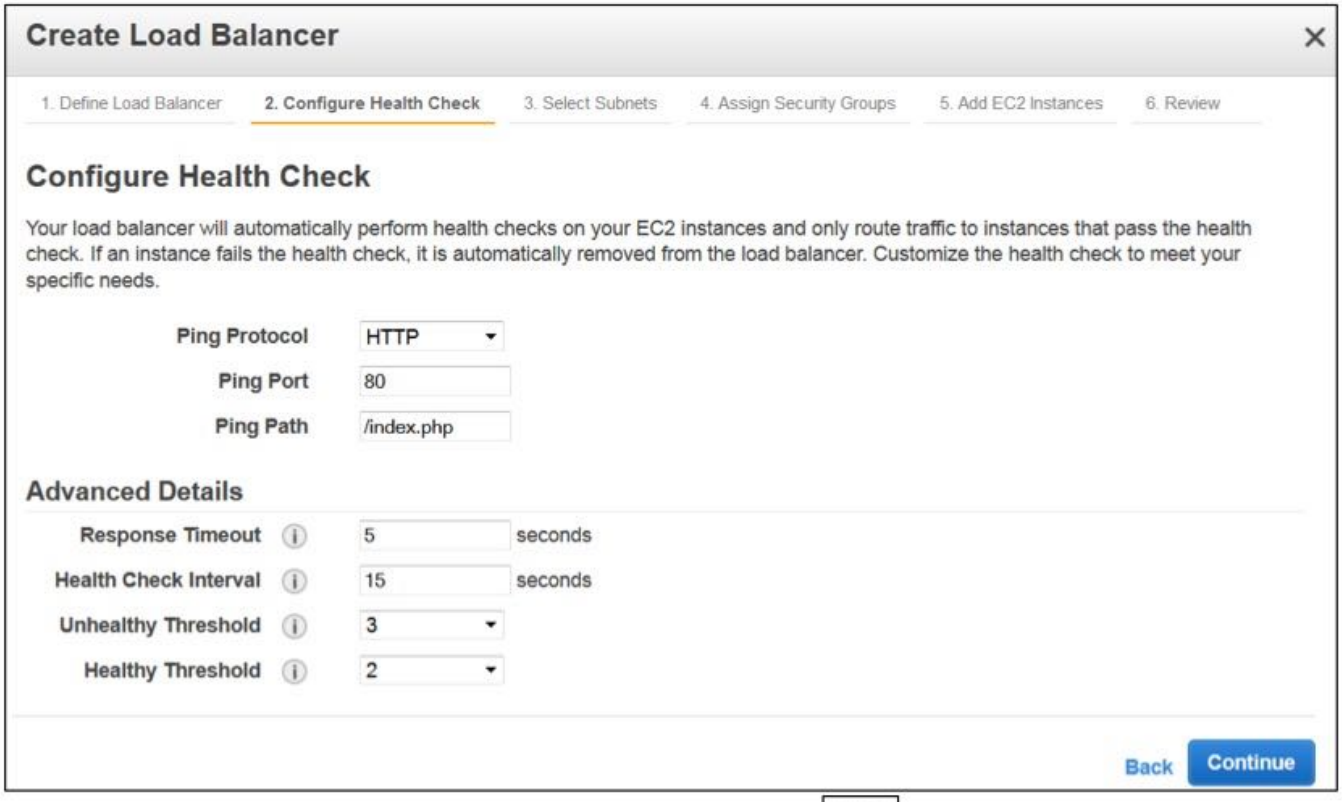
This section of the lab will have you build the Auto Scaling functionality needed to scale up and down the lab web application that will be explored in subsequent labs. Auto Scaling involves several steps, creating a Lunch Configuration, an Auto Scaling group, and one or more Auto Scaling policies. Additionally, Auto Scaling requires an Elastic Load Balancer to create and remove instances from the group.
You can choose to create an ELB and configure your Auto Scaling Group either by using the management console or by using aws cli. In the below section, we will use the management console. If you would like to do the same in CLI, check the Bonus Activity section that is available at the end of this lab.

Creating an Elastic Load Balancer

ELB, or Elastic Load Balancer, works well with the Auto Scaling functionality. ELB obfuscates the servers behind it so that the general web user will not need to know the name of the server they are actually connected to, rather, they'll connect to the ELB. The ELB does not change throughout the scaling process. Here, you will create an ELB for your web servers to Auto Scale.

12. On the management console select **EC2 > Load Balancers > Create Load Balancer**
13. In the **Load Balancer Name** field type **ASLoadBalancer**
14. In the **Create LB inside** field choose your **VPC** and click on **Continue**
15. As displayed in the picture below, enter the following.
    a. **Ping Port** field enter **80.**
    b. **Ping Path** filed enter **/index.php**.
    c. **Response Timeout** field enter **5.**
    d. **Health Check Interval** field enter **15 seconds**.
    e. **Unhealthy Threshold** field enter **3**.
    f. **Healthy Threshold** field choose **2** and click **Continue.**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



16. On the next screen select the two Public Subnets by choosing and click **Continue.**
17. Next choose the **ELBSecurityGroup** and click **Continue.**
18. Make sure you do not select any instance on the next screen as Auto Scaling will add instances behind the ELB later. Click on **Continue**.
19. Finally click on **Create** and click on **Close**.
20. After the ELB is created, you can select the ELB and view its properties.

Creating an Auto Scaling Launch Configuration and Auto Scaling Group

A launch configuration is primarily used to apply specific modifications to instances as they are created programmatically via the Auto Scale up policies. It will consist of configuration that is used to create EC2 instances. That configuration is similar to what you input when creating your instance earlier in this lab.

21. On the management console choose **EC2 > AUTO SCALING > Launch Configurations**.
22. Click on **Create Auto Scaling group**.
23. Select **Create launch configuration**.
24. On the **Create Launch Configuration** screen choose **My AMIs** and **Select** the AMI you created. Select **General purpose** on the left of the screen. Choose **m1.small** on the center of the screen. Select Next**: Configure details**.
25. On the next screen under **Name** enter **ASLaunchConfiguration** and click on **Next: Add Storage**
26. Click on **Next: Configure Security Group.**
27. Choose **Select an existing security group** and select the **WebSecurityGroup** from the list and click on **Review**.
28. Click on **Create launch configuration**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

29. On the Select an existing key pair screen, select **Choose an existing key pair** and select the existing qwikLAB key pair. Choose I acknowledge and click on **Create launch configuration.**
30. Next you will create an Auto Scaling group. This portion will help determine the minimum and maximum number of instances among other details. Do not forget to replace relevant text with values from the lab-details.txt file.
31. On the **Create Auto Scaling Group** screen, enter the following details **Group name** as **WebserversASGroup, Group size** as **2**. **Network** as **VPC** and **Subnet** select both the public subnets. Expand the **Advanced Details** section and select **Receive traffic from Elastic Load Balancer(s)** check box. Further select your **ELB (ASLoadBalancer).** Under **Health Check Type** select **ELB**. Enter **300** under the **Health Check Grace Period** field and select **Next: Configure scaling policies**.
32. On the next screen choose **Use scaling policies to adjust the capacity of this group** and set **Scale between 2** and **4**. This value defines the minimum and maximum number of instances that would be a part of your Auto Scaling group. This group is configured as a 2-4-2 configuration, that is, `minsize-maxsize-desiredcapacity`. In this configuration, a minimum of two instances will be kept up and replaced if health checks fail. Auto Scaling will increase to 4 instances as load increases, and then reduce back down to the desired size of 2 when load decreases.

Configure Auto Scale Policies and Create Alarms

Auto Scale policies configure Auto Scaling to automatically scale based on conditions. Use cases include, CPU, memory or disk thresholds. You will configure both scale up and scale down policies.

33. Configure the scale up policy under the **Increase Group Size** section. Enter a name of your choice.
34. Under the **Execute policy when** section select **Add new alarm**.
35. On the **Create Alarm** screen, uncheck **Send a notification** check box. You can however choose to be notified by creating a topic. Make sure you select Average CPU Utilization to be **>= 50** percent. **For at least:** and **consecutive period(s)** fields must be set to **1 and 1 Minute**.
36. Enter **ASScaleUp** under **Name of Alarm** and click on **Create Alarm**.
37. **Take the action:** should be set to **Add 1** instance. **Add then wait** that refers to a cool down period must be **300**.
38. Similarly under the **Decrease Group Size** section create a scale down policy. Enter a name of your choice.
39. Under the **Execute policy when** section select **Add new alarm**.
40. On the **Create Alarm** screen, uncheck **Send a notification** check box. Like above you can choose to be notified by creating a topic in production. Select Average CPU Utilization to be **<= 35** percent. **For at least:** and **consecutive period(s)** fields must be set to **1 and 1 Minute.**
41. Enter **ASScaleDown** under **Name of Alarm** and click on **Create Alarm**
42. **Take the action:** should be set to **Remove 1** instance. **Add then wait** to **300**.
43. Click on **Next: Configure Notifications**.
44. You can add a notification to be alerted about any change in the Auto Scaling Group.
45. Click on **Review** and verify if you have selected the right options and then click on **Create Auto Scaling group**.
46. Once you receive the **Successfully created Auto Scaling group** message click the **Close** button.
47. You can now select your newly created Auto Scaling group and review its settings.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

48. On the management console choose **EC2 > Instances** and verify that two instances are being created.
49. Find your way to **EC2 > Load Balancers** and click the load balancer you created. Now select the **Instances** tab and look for the two instances that were created by Auto Scaling. Keep refreshing the screen until both the instances are **In Service**. This indicates that the instances are ready to accept web traffic.
50. Under the **Description** tab you should now be able to find a **DNS Name:** `http://asloadbalancer-xxxxxxxxx.xx-xxxx-x.elb.amazonaws.com`. Copy the same and enter it in the address bar of your browser and you should see a web page that is served from one of the backend instances.
51. You can add some CPU load on one of the backend instance to see Auto Scaling scale up. This can be done by appending putcpuload.php at the end of your ELB DNS Name. Click on **Generate Load**. Example `http://asloadbalancer-xxxxxxxxx.xx-xxxx-x.elb.amazonaws.com/putcpuload.php`. It could take few minutes before a new instance is fired up.
52. As per the Auto Scaling policy to scale up, you should now see a new instance being created.
53. On the qwikLAB screen click on **END LAB** to end the lab exercise.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Bonus Activity

Note: Please proceed with the below steps only if you skipped the ELB and Auto Scaling GUI steps.

Creating an Elastic Load Balancer using aws cli

ELB, or Elastic Load Balancer is required for Auto Scaling functionality. ELB obfuscates the servers behind it so that the general web user will not need to know the name of the server they are actually connected to, rather, they'll connect to the ELB. The ELB does not change throughout the scaling process. Here, you will create your ELB for your web servers to reside and Auto Scale behind.

1. The names of your resources you will need can be found on your Bastion Host at `/home/ec2-user/lab-details.txt`

```
aws elb create-load-balancer --load-balancer-name ASLoadBalancer --listeners
"LoadBalancerPort=80,InstancePort=80,Protocol=http,InstanceProtocol=http" --
subnets <PublicSubnet1> <PublicSubnet2> --security-groups <ELBSecurityGroup>
```

   The resulting output will be similar to the following:
   `"DNSName": "ASLoadBalancer-173888080.us-east-1.elb.amazonaws.com"`

2. You will need to configure the health checks:

```
aws elb configure-health-check --load-balancer-name ASLoadBalancer --health-
check
Target=HTTP:80/index.php,Interval=15,Timeout=5,UnhealthyThreshold=3,HealthyTh
reshold=2
```

   Your output will be similar to this:

```
          "Health Check": {
              "HealthyThreshold": 2,
              "Interval": 15,
              "Target": "HTTP:80/index.php",
              "Timeout": 5,
        "UnhealthyThreshold": 3
```

3. Validate your ELB with the following:

```
aws elb describe-load-balancers
```

   The results should be similar to the following:

```
{
    "LoadBalancerDescriptions": [
        {
            "Subnets": [
                "subnet-3bcd8a13",
                "subnet-6dd2e819"
            ],
            "CanonicalHostedZoneNameID": "Z3DZXE0Q79N41H",
            "CanonicalHostedZoneName": "ASLoadBalancer-173888080.us-
east-1.elb.amazonaws.com",
            "ListenerDescriptions": [
                {
                    "Listener": {
                        "InstancePort": 80,
                        "LoadBalancerPort": 80,
                        "Protocol": "HTTP",
                        "InstanceProtocol": "HTTP"
                    },
                    "PolicyNames": []
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
                }
            ],
            "Health Check": {
                "HealthyThreshold": 2,
                "Interval": 15,
                "Target": "HTTP:80/index.php",
                "Timeout": 5,
                "UnhealthyThreshold": 3
            },
            "VPCId": "vpc-7b89691e",
            "BackendServerDescriptions": [],
            "Instances": [],
            "DNSName": "ASLoadBalancer-173888080.us-east-
1.elb.amazonaws.com",
            "SecurityGroups": [
                "sg-269a5b43"
            ],
            "Policies": {
                "LBCookieStickinessPolicies": [],
                "AppCookieStickinessPolicies": [],
                "OtherPolicies": []
            },
            "LoadBalancerName": "ASLoadBalancer",
            "CreatedTime": "2014-01-27T03:47:28.350Z",
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b"
            ],
            "Scheme": "internet-facing",
            "SourceSecurityGroup": {
                "OwnerAlias": "972013970906",
                "GroupName": "qlstack2-lab-135-1390793521-
ELBSecurityGroup-Q95O1B9Y5F5H"
            }
        }
    ]
}
```


Creating an Auto Scaling Launch Configuration and Auto Scaling Group using aws cli

A launch configuration is primarily used to apply specific modifications to instances as they are created programmatically via the Auto Scale up policies. They will consist defaults data that gets put to EC2. That data is similar to what you input when creating your instance earlier in this lab.

4. Input the following to create an Auto Scaling launch configuration, replacing the appropriate fields with your own data. **Note:** the `customAMIid` is the AMI id of the AMI you created.

```
aws autoscaling create-launch-configuration --instance-type m1.small
--key-name <qwikLABS-xxxx-xxx> --launch-configuration-name
ASLaunchConfiguration --image-id <customAMIid> --security-groups
<WebSecurityGroup>
```

With this command `aws autoscaling`, you are enacting the EC2 Auto Scaling service with the operation `create-launch-configuration` and the following parameters:
Image-id – This ID is the ID associated with your custom AMI that you created earlier in the lab.
Instance-type – This is your instance "size", IE: t1.micro, m1.small.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Key – This is the key pair to use.
Launch-configuration-name – This name will be used to refer to the launch configuration.
Security-groups – This can either be the name or the Group ID.
The output will be an empty set, again.

5. Then, double check the launch configuration parameters to verify the command completed with expected results by typing:

```
aws autoscaling describe-launch-configurations
```

This command will display similar resulting information:

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn:aws:autoscaling:us-east-
1:972013970906:launchConfiguration:f04a67b2-d282-405f-b0e2-
4112a4f0fa14:launchConfigurationName/ASLaunchConfiguration",
            "InstanceMonitoring": {
                "Enabled": true
            },
            "ImageId": "ami-05355a6c",
            "CreatedTime": "2014-01-27T04:18:13.156Z",
            "BlockDeviceMappings": [],
            "KeyName": "bastion-qlstack2-lab-135-1390793521",
            "SecurityGroups": [
                "sg-2f9a5b4a"
            ],
            "LaunchConfigurationName": "ASLaunchConfiguration",
            "KernelId": null,
            "RamdiskId": null,
            "InstanceType": "m1.small"
        }
    ]
}
```

Note that you could also inject userdata at this point, in the Auto Scaling launch configuration.

6. Next you will create an Auto Scaling group to determine the resources to be used for the Auto Scaling of the web instances. Again, replace <> text with your values.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name
WebserversASGroup --launch-configuration-name ASLaunchConfiguration --min-
size 2 --max-size 4 --desired-capacity 2 --health-check-type ELB --health-
check-grace-period 300 --load-balancer-name ASLoadBalancer --vpc-zone-
identifier <PublicSubnet1>,<PublicSubnet2>
```

This group is configured as a 2-4-2 configuration, that is, `minsize-maxsize-desiredcapacity`. In this configuration, a minimum of two instances will be kept up and replaced if health checks fail. Auto Scaling will increase to 4 instances as load increases, and then reduce back down to the desired size of 2 when load decreases. The launch-`configuration-name` refers back to the launch configuration you created.

7. Validate your new Auto Scaling group with the following:

```
aws autoscaling describe-auto-scaling-groups
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

The result will be similar to the following:

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-
1:972013970906:autoScalingGroup:3ec12c00-8407-4a2a-8504-
c104fa1a8170:autoScalingGroupName/WebserversASGroup",
            "HealthCheckGracePeriod": 30,
            "SuspendedProcesses": [],
            "DesiredCapacity": 2,
            "Tags": [],
            "EnabledMetrics": [],
            "LoadBalancerNames": [
                "ASLoadBalancer"
            ],
            "AutoScalingGroupName": "WebserversASGroup",
            "DefaultCooldown": 300,
            "MinSize": 2,
            "Instances": [
                {
                    "InstanceId": "i-1b5f173b",
                    "AvailabilityZone": "us-east-1b",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService",
                    "LaunchConfigurationName": "ASLaunchConfiguration"
                },
                {
                    "InstanceId": "i-185f1738",
                    "AvailabilityZone": "us-east-1b",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService",
                    "LaunchConfigurationName": "ASLaunchConfiguration"
                }
            ],
            "MaxSize": 4,
            "VPCZoneIdentifier": "subnet-3acd8a12",
            "TerminationPolicies": [
                "Default"
            ],
            "LaunchConfigurationName": "ASLaunchConfiguration",
            "CreatedTime": "2014-01-27T05:12:58.174Z",
            "AvailabilityZones": [
                "us-east-1b"
            ],
            "HealthCheckType": "ELB"
        }
    ]
}
```

8. Next, you will validate the Auto Scaling instance that is recognized within the Auto Scaling group.

```
aws autoscaling describe-auto-scaling-instances
```

Your output should be similar to:

```
{
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
    "AutoScalingInstances": [
        {
            "AvailabilityZone": "us-east-1c",
            "InstanceId": "i-78018c1c",
            "AutoScalingGroupName": "WebserversASGroup",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService",
            "LaunchConfigurationName": "ASLaunchConfiguration"
        },
        {
            "AvailabilityZone": "us-east-1c",
            "InstanceId": "i-7a018c1e",
            "AutoScalingGroupName": "WebserversASGroup",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService",
            "LaunchConfigurationName": "ASLaunchConfiguration"
        }
    ]
}
```

9. Validate Auto Scaling functionality by terminating an instance (not the BASTION or NAT instance!). A new instance should be recreated. Gather an Instance ID from the previous output, then type:

```
aws ec2 terminate-instances --instance-ids <instance-id>
```

The output should be similar to the following:

```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-7a018c1e",
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "PreviousState": {
                "Code": 48,
                "Name": "running"
            }
        }
    ]
}
```

10. Now, describe the Auto Scaling instances again:

```
aws autoscaling describe-auto-scaling-instances
```

And view that 2 instances are again available. If you do not see 2 instances, wait for a minute or two for the instance to finish its boot process and try again. You could also validate your instances from the management console. Go to **EC2 > Instances > Instances** and validate 2 m1.small instances are now being launched or have been launched.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Configure Auto Scale Policies using aws cli

Auto Scale policies configure Auto Scaling to happen automatically based on conditions. Use cases include, CPU thresholds, memory thresholds and disk thresholds. You will configure both scale up and scale down policies.

11. Return to the terminal window on your Bastion Host to perform these steps.
12. Configure the scale up policy first.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name
WebserversASGroup --policy-name ASscaleUPpolicy --scaling-adjustment
1 --adjustment-type ChangeInCapacity --cooldown 300
```

Results will be an ARN:

```
{
    "PolicyARN": "arn:aws:autoscaling:us-east-
1:972013970906:scalingPolicy:a285e3e1-cbdd-40ca-88d7-
d6f97d2d8824:autoScalingGroupName/WebserversASGroup:policyName/ASscaleU
Ppolicy"
}
```

13. Now, create the scale down policy.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name
WebserversASGroup --policy-name ASscaleDownpolicy --scaling-
adjustment -1 --adjustment-type ChangeInCapacity --cooldown 300
```

Again, the results will be an ARN:

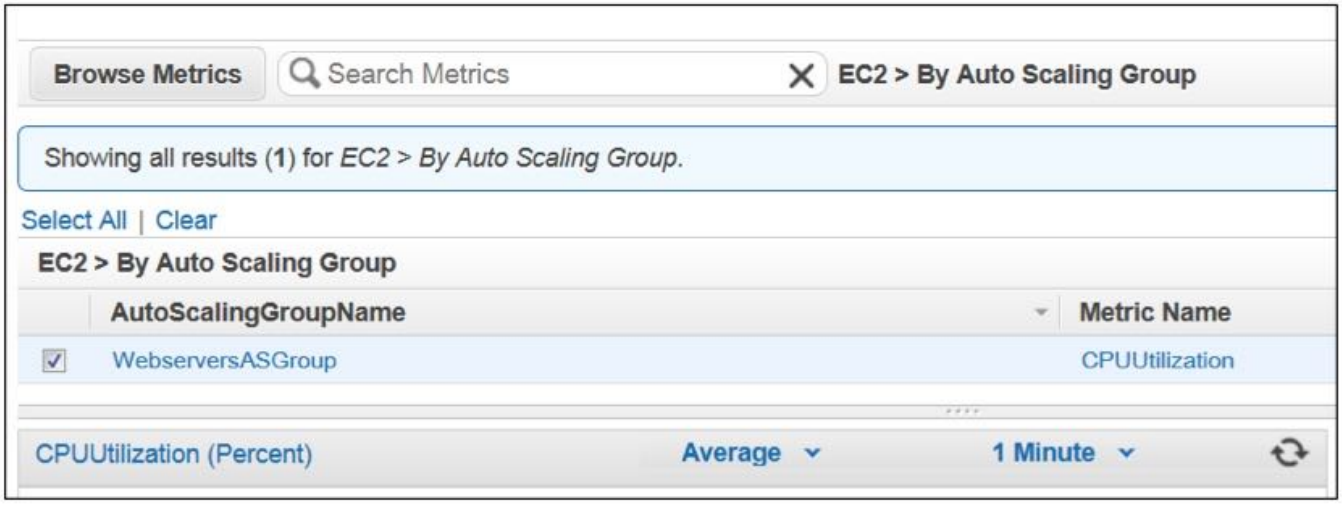
```
{
    "PolicyARN": "arn:aws:autoscaling:us-east-
1:972013970906:scalingPolicy:df5f7b69-bc80-4689-b1ab-
ae5d485c9432:autoScalingGroupName/WebserversASGroup:policyName/ASscaleD
ownpolicy"
}
```

Configure Cloud Watch CPU Alerts

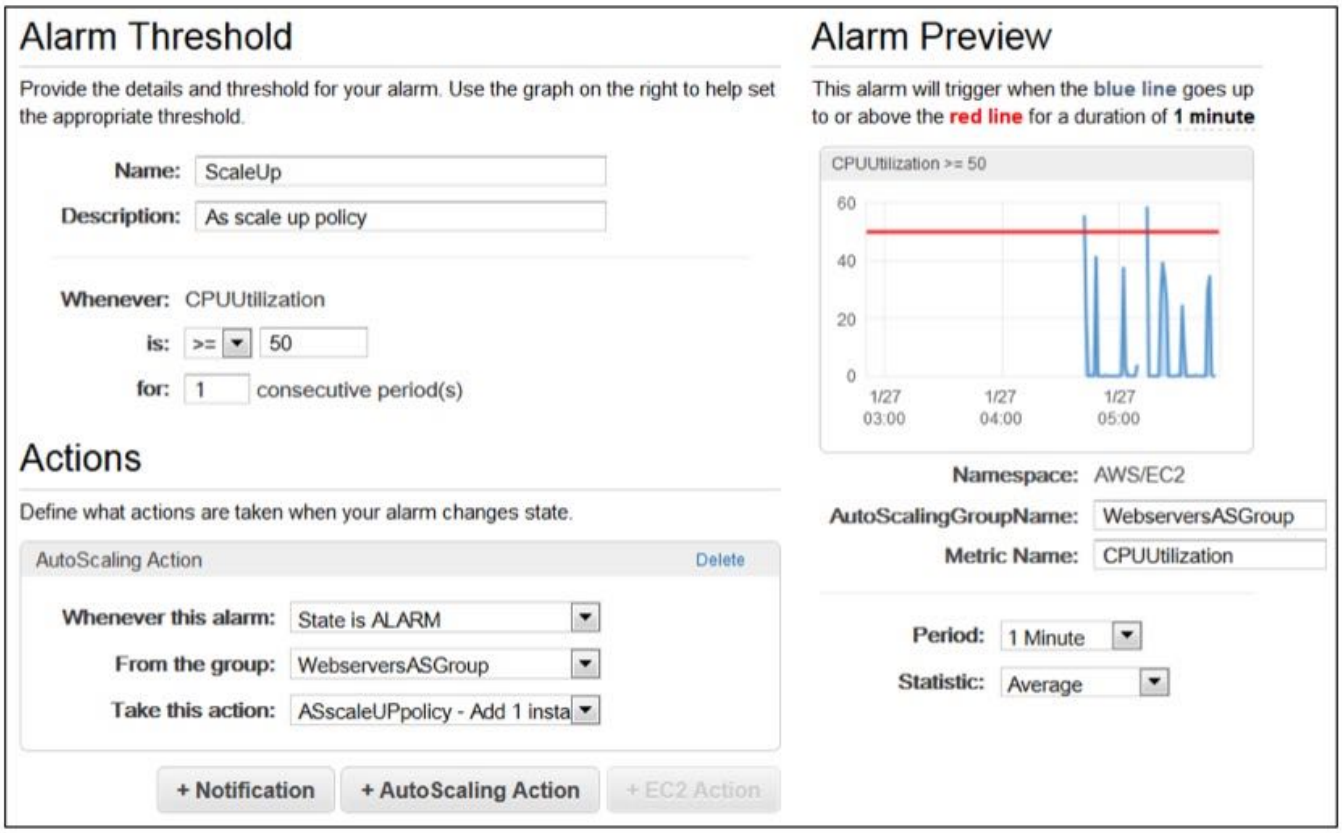
You need to now configure your Cloud Watch alarm to alert Auto Scaling when aggregate CPU is above and when it is below the threshold you set to perform scale up and scale down operations. It takes a few minutes for Cloud Watch to begin gathering statistics and to recognize your new Auto Scaling group.

14. First, create the scale up Cloud Watch alarm.
15. Return to the management console and go to **Services > All AWS Services > Cloud Watch**
16. Click **Alarms**, and then click **Create Alarm**.
17. Search for **By Auto Scaling Group:** and chose the alarm associated with "**WebserversASGroup**" and "**CPUUtilization**", and change the Period from 5 Minutes of **1 Minute**. Then, click **Next.**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



18. Provide a meaningful scale up name and description, and set the utilization to **>= 50**, for **1** minute. Under **Actions > delete** the Notification. Click on **+Auto Scaling Action** > **From the group:** select **WebserversASGroup** and under **Take this action:** select **ASscaleUPpolicy** and click on **Create Alarm**.



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

19. Now, you will need to create the scale down alarm.
20. Click **Alarms**, and then click **Create Alarm**.
21. Search for **By Auto Scaling Group:** and chose the alarm associated with "**WebserversASGroup**" and "**CPUUtilization**", and change the Period from 5 Minutes of **1 Minute**. Then, click **Next**.
22. Provide a meaningful scale down name and description, and set the utilization to **<= 35**, for **1** minute. Under **Actions > delete** the Notification. Click on **+Auto Scaling Action** > **From the group:** select **WebserversASGroup** and under **Take this action:** select **ASscaleDownpolicy** and click on **Create Alarm**.
23. View your newly created Metrics and history by selecting one of your policies at the **Alarms** screen:

24. Find your way to **EC2 > Load Balancers** and click the load balancer you created. Now select the **Instances** tab and look for the two instances that were created by Auto Scaling. Keep refreshing the screen until both the instances are **In Service**. This indicates that the instances are ready to accept web traffic.
25. Under the **Description** tab you should now be able to find a **DNS Name:** `http://asloadbalancer-xxxxxxxxx.xx-xxxx-x.elb.amazonaws.com`. Copy the same and enter it in the address bar of your browser and you should see a web page that is served from one of the backend instances.
26. You can add some CPU load on one of the backend instance to see Auto Scaling scale up. This can be done by appending putcpuload.php at the end of your ELB DNS Name. Example `http://asloadbalancer-xxxxxxxxx.xx-xxxx-x.elb.amazonaws.com/putcpuload.php`.
27. As per the Auto Scaling policy to scale up, you should now see a new instance being created. It could take a while before the instance can start booting.
28. On the qwikLAB screen click on **END LAB** to end the lab exercise.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Conclusion

Congratulations! You now have successfully:

- Bootstrapped an instance from a file
- Repackaging for conversion to an AMI
- Configured an Auto Scaling Launch Configuration
- Configured Auto Scaling Groups
- Configured Auto Scaling Policies
- Configured an Elastic Load Balancer
- Configured Cloud Watch Alarms for Auto Scaling activities

End Your Lab

54. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

55. Close any active SSH client sessions or remote desktop sessions.
56. Click the **End Lab** button on the *qwikLAB*™ lab details page.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

57. When prompted for confirmation, click **OK**.
58. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operation on AWS
Windows Lab Guides

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS Amazon Elastic Compute Cloud (EC2) Windows Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview
In this lab, you will build upon your previous lab experience, and deploy and bootstrap an instance within a VPC. For the most part, you will be using the AWS Command Line Interface (AWS CLI) to interact with EC2. This lab will help reinforce the understanding of EC2 relating to system usage, maintenance and management. You will learn techniques that can be helpful when creating *scripts* to automate a wide range of tasks. This is a fundamental lab, which other labs will build off of.

Objectives
This lab is intended to build off of a previous lab involving Amazon VPC and prepares you to further extend the infrastructure for deploying a web application.

Technical Knowledge Prerequisites
To successfully complete this lab, you should be familiar with basic Windows server administration and comfortable working with a command-line interface. The ability to navigate directories at the command-line, install tools, and manage volumes will be required; however, step-by-step instructions are provided to assist you.

Topics Covered
The following topics will be covered:

- Using a command-line interface to configure an EC2 instance
- Creating Key Pairs
- Creating and configuring Security Groups
- Launching EC2 Windows Instances
- Describing and Tagging resources
- Connecting to EC2 Windows Instances

Amazon EC2 Instance Deployment

Process Overview
In this section, you will launch an EC2 Instance that will be a Database server. But before effectively launching the instance, a few steps are required.
All the steps and procedures described below could be done through the AWS Management Console. However, in this course, we focus on the AWS CLI. For this reason, we are going to do almost everything from the command line. Learning how to use the AWS CLI can be very helpful for you in the future, when you will want to write scripts to help you with management and automation of your own infrastructure.
The following steps describe the whole process:

11. Connect to the Bastion Host [AWS Management Console]
12. Inspect the Instance Metadata
13. Inspect the AWS CLI configuration and documentation [CLI]
14. Create a Security Group [CLI]
15. Create a Key Pair [CLI]
16. Launch the EC2 instance [CLI]
17. Connect to the instance and configure it [CLI]

Connect to the Bastion Host
All the steps prior to launching the EC2 Instance will be executed on the Bastion Host that you learned how to create on a previous lab. The benefits of using a Bastion Host include:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

- **Increased security**: by restricting access to your other resources to the Bastion Host only, there's less surface for attackers to hack into your infrastructure.
- **Auditing**: you could set up logging and auditing on the Bastion Host, so that every modification to your infrastructure would be securely logged and traceable. [*not covered on this lab*]
- **Easier to interact with the Services:** by using a Bastion Host for each infrastructure you manage on AWS, you don't need to rely on management software and settings being deployed on your own computer, on each of your company's computers, etc. All the tools are installed and configured in a central place – the Bastion Host – providing you with a homogeneous management environment that can easily be replicated.

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair

This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this guide called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group

It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

18. Retrieve your current public IP address by navigating your browser to http://ip.appspot.com
19. Open the **AWS Management Console**.
20. Select the **VPC** service.
21. From the menu on the left side, click **Security Groups**.
22. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion Host"**.
23. On the bottom pane, click the **Inbound** tab.
24. From the **Create a new rule dropdown**, select RDP.
25. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
26. Click the **Add rule** button.
27. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host

28. Open the **AWS Management console**.
29. Select the **EC2** service.
30. From the menu on the left side, click **Instances**.
31. **Select the BastionHost instance** by clicking on the box next to its name.
32. The Public IP is displayed on the bottom pane. Please write it down.

Retrieve the bastion instance password

Before connecting to the bastion using RDP, you must retrieve the instance's password for the Windows Administrator account.

33. In your browser, switch to the **qwikLAB** tab.
34. In the **Lab Connection** portion of the page, copy the **Password** in the **RDP** section – you may also paste it into a text file for later retrieval.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: This is the password you use to log into the bastion instance using the remote desktop client.

35. If the computer you are using to complete this lab is Windows-based, proceed to the next section. If you are using an OS X or Linux-based machine, proceed to the section appropriate to your operating system.

Connecting to the bastion instance (Windows)

Important: Proceed to one of the later sections entitled "Connect to the bastion instance (OS X)" or "Connect to the bastion instance (Linux)" if you are using a Linux-based laptop to perform this exercise. This section is *only* for Microsoft Windows users. There is no need for you to perform the steps in all sections. Only complete the section that matches your operating system.

36. On your local computer click **Start > Run**, type **MSTSC** and click **OK** to start the local RDP client.
37. Type or paste the Elastic IP address of the bastion instance in the **Computer** field.
38. Click **Show Options**, and type **\Administrator** in the **User name** field.

Note: Specifying the user name as "\Administrator" (with a leading backslash) differentiates the bastion Administrator account from the Administrator account on your local computer.

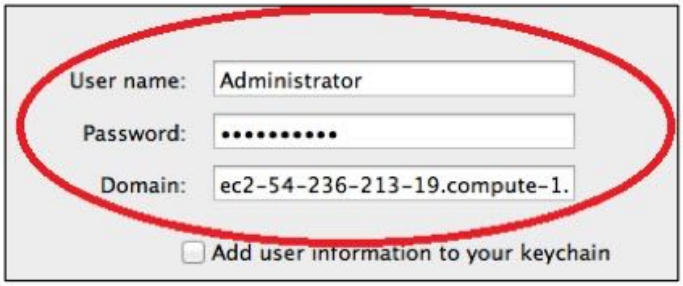
39. Click **Connect**.
40. When prompted, type the RDP **Password** you noted then click **OK**.
41. Click **Yes** if you see a certificate verification message similar to "the identity of the remote computer cannot be verified:"

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

42. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (OS X)
Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." This section is for OS X users *only*. If you are using a different Linux-based operating system, please proceed to the section entitled "Connect to the bastion instance (Linux)." There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

43. Open the Remote Desktop Connection for Mac application.
44. Type the bastion instance's IP address you noted in the **Computer** field and click **Connect**.
45. When prompted, in the **User name** field, type Administrator and then type the **Password** you noted from the qwikLAB page. The Domain should be automatically set to the EC2 Instance DNS name, and you can ignore it.



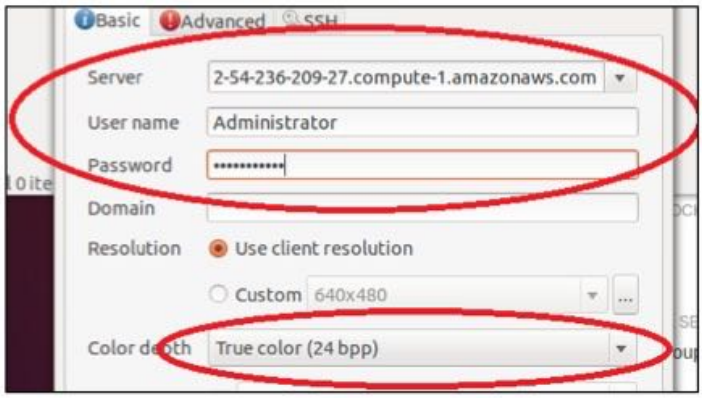
46. Click **OK**.
47. Click **Connect** if you see a verification message similar to "the server name is incorrect."
48. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (Linux)
Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." Only complete this section if you are using a non-

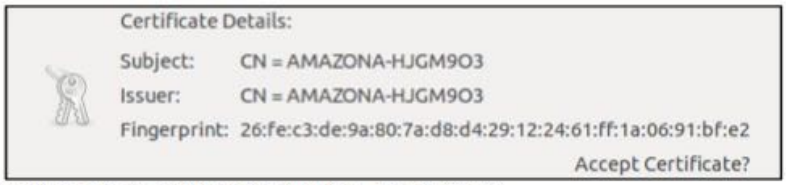
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Macintosh, Linux-based laptop to perform this exercise. This section is *only* for Linux users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

49. Open the Remmina Remote Desktop Client.
50. Type the bastion instance's IP address you noted in the **Server** field.
51. In the **User name** field type **Administrator**, and in the **Password** field, type the password from the qwikLAB page.
52. Optionally, choose a **Color depth** that your bandwidth supports (in this example 'True color (24 bpp)') for a nicer remote desktop environment.



53. Click **Connect**.
54. Click **OK** when prompted to accept the remote certificate.



55. Proceed to the next section entitled "Using the AWS CLI."

Inspect the Instance Metadata

For the next section, you will collect information about the Bastion Host and its running environment (e.g. region, VPC ID, and security group id). There are multiple ways to obtain it. In this lab, we are going to retrieve all that information by querying the Instance Metadata service.

Explore the Metadata service

Every EC2 instance has a Metadata service handled by the physical host's hypervisor that can be queried by issuing HTTP GET requests to the endpoint http://169.254.169.254/. You are going to use the `curl` tool to explore the Metadata service. This tool is installed for you automatically by the lab environment. To use it on your own servers, you will need to install it. It can be found at http://curl.haxx.se/.

66. Open a command shell by typing the following command at the Windows Start screen, followed by the ENTER key.

```
cmd
```

67. At the command prompt, run the command:

```
curl -s -w \n http://169.254.169.254/
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

68. Observe the list of versions of the service.
69. Obtain information on the `latest` version:

```
curl -s -w \n http://169.254.169.254/latest/
```

70. Observe the available categories of data, including `meta-data`.
71. List all the Metadata options available:

```
curl -s -w \n http://169.254.169.254/latest/meta-data/
```

72. Inspect the `placement` related data:

```
Curl -s -w \n http://169.254.169.254/latest/meta-data/placement/
```

Retrieve the Region

73. Retrieve the instance's availability zone:

```
curl -s -w \n http://169.254.169.254/latest/meta-
data/placement/availability-zone
```

74. Take note of the Region of the Bastion Host, by discarding the last letter of its Availability Zone.

Retrieve the VPC ID

75. List the instance's elastic network interfaces:

```
curl -s -w \n http://169.254.169.254/latest/meta-
data/network/interfaces/macs/
```

76. Save the MAC address to an environment variable called BASTION_MAC:

```
set command=curl -s -w \n http://169.254.169.254/latest/meta-
data/network/interfaces/macs/
for /f "delims=" %A in ('%command%') do @set BASTION_MAC=%A
```

77. List the available network information:

```
curl -s -w \n http://169.254.169.254/latest/meta-
data/network/interfaces/macs/%BASTION_MAC%
```

78. Save the VPC ID to an environment variable called BASTION_VPC:

```
set command=curl -s -w \n http://169.254.169.254/latest/meta-
data/network/interfaces/macs/%BASTION_MAC%vpc-id
for /f "delims=" %A in ('%command%') do @set BASTION_VPC=%A
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Retrieve the Security Group ID

79. Retrieve the ID of the Bastion Host's security group :

```
curl -s -w \n http://169.254.169.254/latest/meta-
data/network/interfaces/macs/%BASTION_MAC%security-group-ids
```

80. Save the name of the Bastion Host's security group to an environment variable called BASTION_SG:

```
set command=curl -s -w \n http://169.254.169.254/latest/meta-
data/network/interfaces/macs/%BASTION_MAC%security-group-ids
for /f "delims=" %A in ('%command%') do @set BASTION_SG=%A
```

81. Verify that the environment variable has been correctly initialized:

```
set bastion
```

Retrieve the AMI ID

The Bastion Host has been launched with an Amazon Linux AMI image. During this lab, you are going to need the same AMI ID to launch a new instance. For this reason, you are going to retrieve the AMI ID from the Instance Metadata.

82. Save the Bastion Host's AMI ID to an environment variable called AMI_ID:

```
set command=curl -s -w \n http://169.254.169.254/latest/meta-
data/ami-id
for /f "delims=" %A in ('%command%') do @set AMI_ID=%A
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Inspect the AWS CLI configuration and documentation

Change and inspect settings

Before you begin using the AWS CLI, you will change a setting that controls the format of the output of the commands. By default, the commands will output JSON, which is not the best format to interact with other command line tools.

83. From the EC2 Instance prompt, **run the following command**:

```
aws configure
```

84. Leave the Access Key blank and **hit enter**.
85. Leave the Secret Key blank and **hit enter**.
86. Specify the default region name as the Bastion Host's region obtained earlier and **hit enter**.
87. Specify the default output format: **text**, and **hit enter**.

[optional] Obtaining help

The AWS CLI tools comes with extensive documentation. To read the manual, you just need to end any command with the argument `help`.

Let's explore the documentation with the following steps:

88. Run the following command:

```
aws help
```

89. Scroll up and down **using the arrow keys** on your keyboard.
90. When you are done, return to the prompt by pressing the key **q** on your keyboard.

The previous steps will show you general information on the AWS CLI, including a list of supported services. You can obtain help specific to each supported service and each supported API.

91. Obtain help on the EC2 service:

```
aws ec2 help
```

92. When you are done, return to the prompt by pressing the key **q** on your keyboard.
93. Obtain help on the RunInstances API of the EC2 service:

```
aws ec2 run-instances help
```

Create a Security Group

As you should know by now, it is recommended to have a separate Security Group for each logical grouping of servers you maintain (for example, one for the web-servers, another one for the application servers, another one for a specific background worker, another one for your database layer, and so on). For this reason, you are going to create a specific Security Group for the Database server you are going to launch. The name of the security group will be *db-sg*. For now, it will allow ingress RDP traffic from the Bastion Host only, so that you will be able to connect to it and configure it.

94. Create the security group by running the following command, and save its ID to an environment variable called DB_SG:

```
set command=aws ec2 create-security-group --group-name db-sg --vpc-
id %BASTION_VPC% --description database-security-group --query
GroupId
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
for /f "delims=" %A in ('%command%') do @set DB_SG=%A
```

95. Authorize ingress RDP from the Bastion Host security group:

```
aws ec2 authorize-security-group-ingress --group-id %DB_SG% --
protocol tcp --port 3389 --source-group %BASTION_SG%
```

Create a Key Pair

Security best practice state to have a different Key Pair for each logical grouping of servers you maintain. For this reason, you are now going to create a Key Pair for your Database server.

Create the Key Pair

96. Run the following command:

```
aws ec2 create-key-pair --key-name db-kp --query
KeyMaterial > %userprofile%\db-kp.pem
```

The file that gets created is your private key. You will need this key in order to decrypt the Windows administrator password EC2 instances created using the associated public key.

Launch the EC2 instance

The last piece of information needed to launch the EC2 instance is the Private Subnet id. After obtaining this information, you are going to launch the instance and save its ID to an environment variable called DB_INSTANCE_ID which will be used later.

Retrieve the Private Subnet id

When the lab was created, all the resources were tagged. For example, the Subnets tags should read Network=Public and Network=Private. We will look for these tags to determine the private subnet ID. This should show you the importance of properly tagging your resources!

97. Save the private subnet ID to an environment variable called PRIV_SUBNET:

```
set command=aws ec2 describe-subnets --filters
Name=tag:Network,Values=Private --query Subnets[*].SubnetId
for /f "delims=" %A in ('%command%') do @set PRIV_SUBNET=%A
```

Launching the EC2 instance

98. Create the instance:

```
set command=aws ec2 run-instances --image-id %AMI_ID% --key-name db-
kp --security-group-ids %DB_SG% --instance-type m1.small --subnet-
id %PRIV_SUBNET% --query Instances[*].InstanceId
for /f "delims=" %A in ('%command%') do @set DB_INSTANCE_ID=%A
```

99. **Tag the instance**:

```
aws ec2 create-tags --resources %DB_INSTANCE_ID% --tags
Key=Name,Value=DB1
```

Verifying the EC2 instance state

100. **Run the following command**:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
aws ec2 describe-instances --instance-ids %DB_INSTANCE_ID%
```

As you can see, there is a lot of information you can obtain from this command. You can use the --query option to retrieve only the state:

101. **Run the following command**:

```
aws ec2 describe-instances --instance-ids %DB_INSTANCE_ID% --query
Reservations[*].Instances[*].State.Name
```

102. **Repeat the previous command** until the instance is in the *running* state.

This is particularly useful when you are creating a script that must wait for an instance to be in the RUNNING state before it can proceed. Just have the script run the previous command (with the --query) repeatedly until a condition is met, then your script can proceed.

Connect to the instance and configure it

Now that the instance is running you can use the AWS Console (on the Bastion Host) to decrypt the Windows administrator password and download the RDP connection shortcut which will allow you to administer the database server from the Bastion Host. You will walk through this remote administration process in a future lab.

Congratulations! You have completed this lab. You should now be able to:

- Use a command-line interface to configure an EC2 instance
- Create Key Pairs
- Create and configure Security Groups
- Launch EC2 Windows Instances
- Describe and Tag resources
- Connect to EC2 Windows Instances

End Your Lab

103. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

104. Close any active SSH client sessions or remote desktop sessions.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

105. Click the **End Lab** button on the *qwikLAB*™ lab details page.

When prompted for confirmation, click **OK**.

106. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
Amazon Elastic Block Store (EBS)
Windows Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

This lab will provide a walkthrough of many common systems operations activities using the robust and exciting toolsets provided to AWS customers. It will also explain how to create EBS volumes for your AWS EC2 instances. The lab is intended to build the basic skills used in the AWS certification program.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with basic Windows server administration and comfortable working with a command-line interface. The ability to navigate directories at the command-line, install tools, and manage volumes will be required; however, steps are provided to assist you.

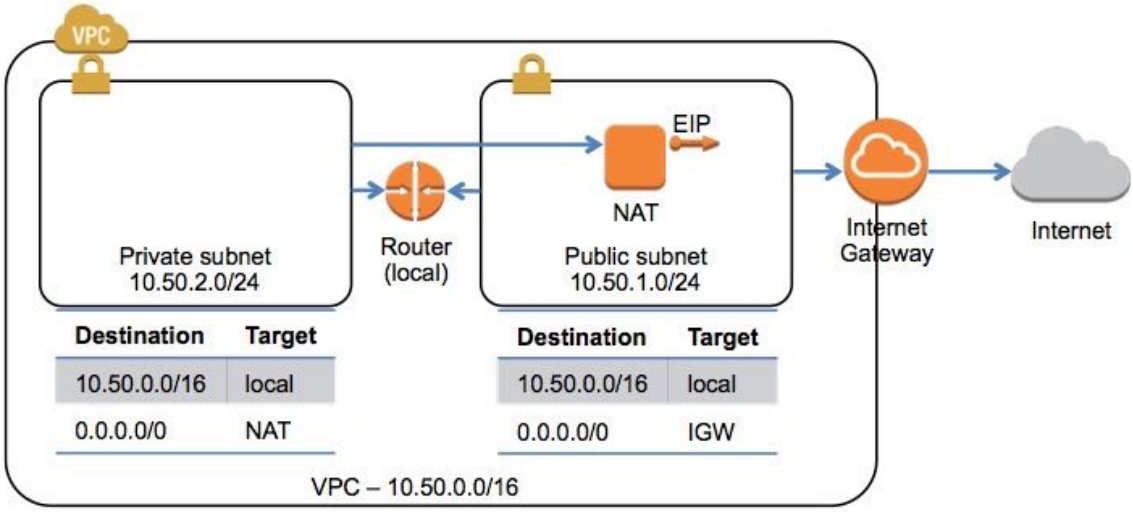
Lab Objectives

After completing this lab you should be able to:

- Use a command-line interface to configure an EC2 instance
- Connect to EC2 instances
- Create EBS volumes
- Create and instantiate snapshots
- Stripe volumes
- Create EBS provisioned IOPS volumes
- Use the AWS command-line tool

Environment

The VPC environment for this lab is pre-configured using a CloudFormation template. The environment looks like the following:



In each of your two Availability Zones, you have a public subnet and two private subnets. The public subnet contains Internet-facing instances such as your NAT instance and your bastion instance. For simplicity (and to reduce the time required to create the lab resources), you have one NAT instance and one bastion host in the 10.50.1.0 public subnet. In practice, these instances would be mirrored across both public subnets in the two AZs for high availability.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

The two private subnets in your VPC are used for other resource tiers that do not need direct Internet access such as your database tier. When these instances need to access the Internet, they communicate through the NAT instance in your public subnet(s). The bastion instance is used in this lab as a publicly accessible jump point that you can use to reach instances in the private subnets (any publicly accessible instance with the AWS command-line tools installed can be used in this manner). Using a pre-configured security group, you can access the bastion instance (and other instances using the security group) via RDP over port 3389.

Part 1: Using the AWS CLI Tools

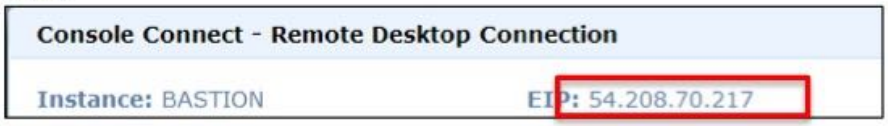
In this section, you run commands from the bastion instance to modify the configuration of the database instance. The bastion instance is used in this lab as a publicly accessible jump point that you use to reach instances in the private subnets (any publicly accessible instance with the AWS command-line tools installed can be used in this manner). The pre-configured security group (BastionHostSecurityGroup), allows access to the bastion instance using RDP over port 3389.

The AWS command-line tools are available for download here: http://aws.amazon.com/cli. Your bastion instance is pre-configured with the CLI tools you need for this lab.

Retrieve the bastion instance IP address and password

Before connecting to the bastion using RDP, you must retrieve the instance's IP address and the password for the Windows Administrator account.

6. Switch to the **EC2 Management Console**.
7. In the **Instances** section, click the **Instances** link.
8. In the AWS Management Console, click the **BASTION** instance.
9. Record the **Elastic IP Address** (EIP) for the instance in a text file (such as a Notepad document).



10. In your browser, switch to the **qwikLAB** tab.
11. In the **Lab Connection** portion of the page, copy the **Password** in the **RDP** section – you may also paste it into a text file for later retrieval.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: This is the password you use to log into the bastion instance using the remote desktop client.

12. If the computer you are using to complete this lab is Windows-based, proceed to the next section. If you are using an OS X or Linux-based machine, proceed to the section appropriate to your operating system.

Connecting to the bastion instance (Windows)

Important: Proceed to one of the later sections entitled "Connect to the bastion instance (OS X)" or "Connect to the bastion instance (Linux)" if you are using a Linux-based laptop to perform this exercise. This section is *only* for Microsoft Windows users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

13. On your local computer click **Start > Run**, type **MSTSC** and click **OK** to start the local RDP client.
14. Type or paste the Elastic IP address of the bastion instance in the **Computer** field.
15. Click **Show Options**, and type **\Administrator** in the **User name** field.

Note: Specifying the user name as "\Administrator" (with a leading backslash) differentiates the bastion Administrator account from the Administrator account on your local computer.

16. Click **Connect**.
17. When prompted, type the RDP **Password** you noted then click **OK**.
18. Click **Yes** if you see a certificate verification message similar to "the identity of the remote computer cannot be verified:"

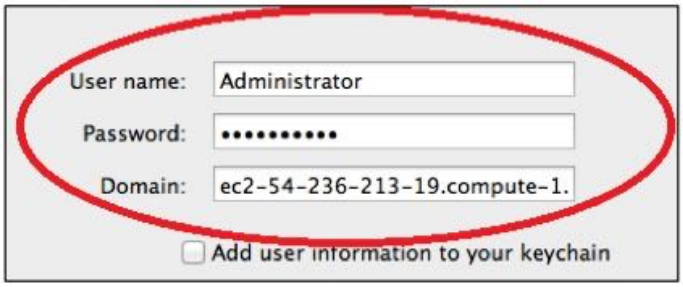
19. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (OS X)

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." This section is for OS X users *only*. If you are using a different Linux-based operating system, please proceed to the section entitled "Connect to the bastion instance (Linux)." There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

20. Open the Remote Desktop Connection for Mac application.
21. Type the bastion instance's IP address you noted in the **Computer** field and click **Connect**.
22. When prompted, in the **User name** field, type Administrator and then type the **Password** you noted from the qwikLAB page. The Domain should be automatically set to the EC2 Instance DNS name, and you can ignore it.



23. Click **OK**.
24. Click **Connect** if you see a verification message similar to "the server name is incorrect."
25. Proceed to the section entitled "Using the AWS CLI."

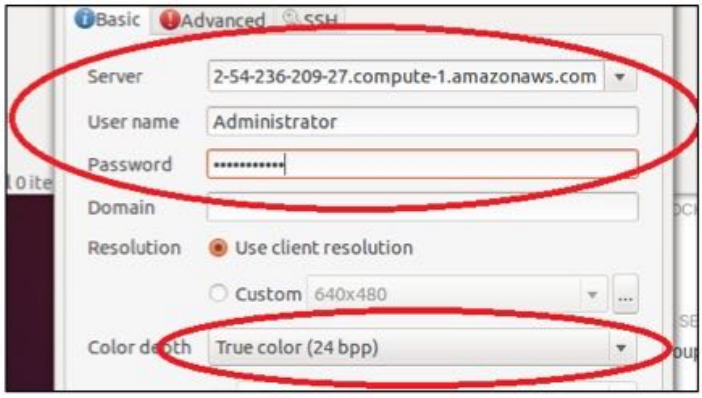
Connecting to the bastion instance (Linux)

Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." Only complete this section if you are using a non-Macintosh, Linux-based laptop to perform this exercise. This section is *only* for Linux users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

29. Open the Remmina Remote Desktop Client.
30. Type the bastion instance's IP address you noted in the **Server** field.
31. In the **User name** field type **Administrator**, and in the **Password** field, type the password from the qwikLAB page.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

32. Optionally, choose a **Color depth** that your bandwidth supports (in this example 'True color (24 bpp)') for a nicer remote desktop environment.



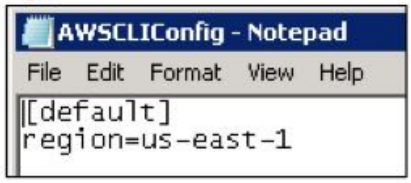
33. Click **Connect**.
34. Click **OK** when prompted to accept the remote certificate.

35. Proceed to the next section entitled "Using the AWS CLI."

Using the AWS CLI

Before using the CLI tools on the bastion instance, you verify the working directory (**`C:\aws`**) that stores your AWS configuration and helpful scripts for this lab. The steps in this section assume a working knowledge of Microsoft Windows and of using a Windows Command Prompt.

36. Using the remote desktop client, verify you have a folder at the root of the bastion instance's **`C:\`** drive named **`\aws`**.
37. In the **`C:\aws`** folder, open **`AWSCLIConfig.txt`**.
38. The file should set the appropriate region for your environment. This file is generated for you during the instance bootstrapping process.



Note: The AWSCLIConfig file supports additional parameters as in the following example:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
[default]
aws_access_key_id=[YourAccessKey]
aws_secret_access_key=[YourSecretAccessKey]
region=[YourRegion]
output=text
```

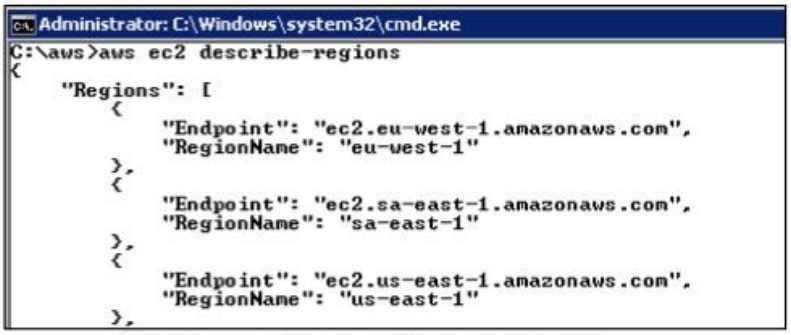
This lab relies on an IAM role for identity management (a preferred best practice) so the aws_access_key_id and aws_secret_access_key parameters are not required. The output=text option (tabular output) is also not required since this lab uses JSON.

39. Close the file.
40. In your remote desktop client, from the **Start** menu, choose **Control Panel**.
41. Click the **System and Security** link.
42. Click the **System** link.
43. Click **Advanced system settings**.
44. On the **Advanced** tab, click **Environment Variables**.
45. Verify you have a Windows environment variable named **AWS_CONFIG_FILE** that points to your AWSCLIConfig file: `C:\aws\AWSCLIConfig.txt`.

46. Close the System Properties and the Control Panel.
47. Open a Windows **Command Prompt** (**Start > Run > cmd**).
48. Test your environment by typing this command to retrieve a list of available EC2 regions:

```
aws ec2 describe-regions
```

Your output should look like the following:



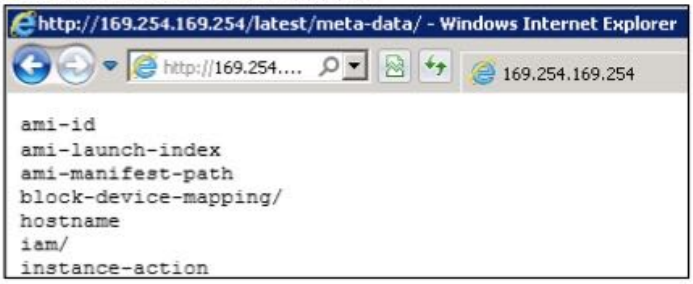
49. Leave the command window open (you use it later in this lab).

Examining EC2 Metadata

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

EC2 instance metadata is a useful way of exposing EC2 instance properties to applications or services that need access to EC2 instance context. In this section of the lab, you examine some of the metadata for your bastion instance. For more information on retrieving instance metadata, see the AWS product documentation: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AESDG-chapter-instancedata.html

50. In your remote desktop client, open a browser window.
51. Paste the following URL into the address bar: **http://169.254.169.254/latest/meta-data**. You should see the available metadata details returned.



52. Each of the metadata properties can be appended to the URL to reveal specific properties for your instance. Paste the following URLs into your browser's address bar and examine the results:

    http://169.254.169.254/latest/**meta-data/instance-id**
    http://169.254.169.254/latest/**meta-data/placement/availability-zone**
    http://169.254.169.254/latest/**meta-data/hostname**
    http://169.254.169.254/latest/**meta-data/security-groups**
53. Leave your remote desktop client running. If the connection is ended by the instance, switching to the remote desktop window will reconnect you to the bastion instance.

Summary: Part 1

The EC2 commands you execute using the AWS CLI return JSON-formatted responses. Using the describe-regions command, for example, provides a list or regions for your EC2 instances including: EU (European Union), SA (South America), US (United States), and AP (Asia Pacific), among others. This list does not include the GovCloud region unless your AWS account is GovCloud enabled. For more details on the GovCloud region, see: http://aws.amazon.com/govcloud-us.

For more information on AWS CLI features, see: http://aws.amazon.com/cli.

For more information on command-line input parameters, see: http://docs.aws.amazon.com/cli/latest/userguide/cli-using-param-commandline.html.

For more information on running PowerShell-equivalent cmdlets on Windows, see the appendixes.

Part 2: Resizing a Boot Disk

Part 2 of this lab requires the instance ID of your storage instance. If you did not record the instance ID previously, in the EC2 Management Console, choose **Instances > Instances**, select the **STORAGE**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

instance and note the instance ID. In this section, you create a snapshot of the boot disk on the storage instance and you attach a new, larger volume to the instance to replace it.

Resizing a boot disk

53. In the remote desktop client, in a Command Prompt, type the following command (replacing [YourInstanceId] with the id of your storage instance):

```
aws ec2 describe-instances --instance-ids [YourInstanceId]
```

54. Scroll through the results and examine the properties of the attached EBS volume.

```
    },
    "Hypervisor": "xen",
    "BlockDeviceMappings": [
        {
            "DeviceName": "/dev/sda1",
            "Ebs": {
                "Status": "attached",
                "DeleteOnTermination": true,
                "VolumeId": "vol-fd69878a",
                "AttachTime": "2013-08-29T18:04:31.000Z"
            }
```

Record the **VolumeId** value. In later steps you use the **VolumeId** to resize the volume.

55. Type the following command to retrieve information about the volume (replacing [YourVolumeId] with the id of your volume):

```
aws ec2 describe-volumes --volume-ids [YourVolumeId]
```

Notice the volume State is "attached" and the Size is 30 (GB).

56. Type the following command to initiate a snapshot of the volume (replacing [YourVolumeId] with the id of your volume).

```
aws ec2 create-snapshot --volume-id [YourVolumeId]
```

Record the **SnapshotId** value. You use it in later steps:

```
\aws>aws ec2 create-snapshot --volume-id vol-fd69878a

  "Description": null,
  "Tags": [],
  "VolumeId": "vol-fd69878a",
  "State": "pending",
  "VolumeSize": 30,
  "Progress": null,
  "StartTime": "2013-08-30T01:54:47.000Z",
  "SnapshotId": "snap-2f4e5f23",
  "OwnerId": "407829951608"
```

Note: Snapshots can be taken whether or not the instance is running. If you create a snapshot while the instance is running, it takes a more time to complete. Also note: Snapshots are created at the region level.

57. View the snapshot's progress by typing the following command (replacing [YourSnapshotId] with the id of your snapshot):

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
aws ec2 describe-snapshots --snapshot-ids [YourSnapshotId]
```

58. The snapshot State is "pending" while the snapshot is in progress. Run the previous command periodically until the State is "completed."

```
\aws>aws ec2 describe-snapshots --snapshot-ids snap-2f4e5f23

  "Snapshots": [
      {
          "Description": null,
          "VolumeId": "vol-fd69878a",
          "State": "completed",
          "VolumeSize": 30,
          "Progress": "100%",
          "StartTime": "2013-08-30T01:54:47.000Z",
          "SnapshotId": "snap-2f4e5f23",
          "OwnerId": "407829951608"
```

59. In order to attach a volume to an instance, the volume must exist in the same AZ as the EC2 instance to which it will be attached. Once a volume is created, it cannot be moved to another AZ. You could, however, create the volume in another AZ using a snapshot. Type the following command to create a larger, 120GB volume from the snapshot (replacing [YourSnapshotId] with the snapshot id and replacing [YourAZ] with the Availability Zone of the storage instance and the bastion):

```
aws ec2 create-volume --snapshot-id [YourSnapshotId] --
size 120 --availability-zone [YourAZ]
```

```
C:\aws>aws ec2 create-volume --snapshot-id snap-0a15fa05 --size 40 --availabilit
y-zone us-east-1b
{
    "AvailabilityZone": "us-east-1b",
    "Attachments": [],
    "Tags": [],
    "VolumeType": "standard",
    "VolumeId": "vol-cc31d9bb",
    "State": "creating",
    "SnapshotId": "snap-0a15fa05",
    "CreateTime": "2013-08-30T18:15:30.616Z",
    "Size": 40
```

Note: If you did not record the AZ of the instances earlier, it is visible on the desktop background of your bastion host instance.

60. Record the **VolumeId** returned by the command. You use it in subsequent steps.

61. Type the following command to view the status of the volume (replacing [YourVolumeId] with the volume id):

```
aws ec2 describe-volumes --volume-ids [YourVolumeId]
```

62. Wait until the **State** of the volume is **available**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
\aws>aws ec2 describe-volumes --volume-ids vol-86df36f1

  "Volumes": [
      {
          "AvailabilityZone": "us-east-1b",
          "Attachments": [],
          "VolumeType": "standard",
          "VolumeId": "vol-86df36f1",
          "State": "available",
          "SnapshotId": "snap-2f4e5f23",
          "CreateTime": "2013-08-30T02:02:29.000Z",
          "Size": 40
```

63. Type the following commands to stop the storage instance and detach the old, smaller volume from it (replacing [YourInstanceId] with the storage instance id and replacing [YourOldVolumeId] with the id of the original, smaller volume):

```
aws ec2 stop-instances --instance-id [YourInstanceId]
aws ec2 detach-volume --volume-id [YourOldVolumeId]
```

After detaching it, the State should change to "detaching."

```
C:\Users\Administrator>aws ec2 detach-volume --volume-id vol-7783b63d
{
    "AttachTime": "2013-08-30T20:31:38.000Z",
    "InstanceId": "i-526b343b",
    "VolumeId": "vol-7783b63d",
    "State": "detaching",
    "Device": "/dev/sda1"
```

64. Type the following command to verify the State of the old volume is "available" to ensure detaching completed (replacing [YourOldVolumeId] with the id of the original, smaller volume):

```
aws ec2 describe-volumes --volume-ids [YourOldVolumeId]
```

```
C:\Users\Administrator>aws ec2 describe-volumes --volume-id vol-7783b63d
{
    "Volumes": [
        {
            "AvailabilityZone": "us-east-1c",
            "Attachments": [],
            "VolumeType": "standard",
            "VolumeId": "vol-7783b63d",
            "State": "available",
            "SnapshotId": "snap-772a2627",
            "CreateTime": "2013-08-30T20:31:37.000Z",
            "Size": 35
```

65. When the old volume is detached, type the following command to attach the new, larger volume (replacing [YourNewVolumeId] with the id of the new, larger volume and replacing [YourInstanceId] with the id of the storage instance):

```
aws ec2 attach-volume --volume-id [YourNewVolumeId] --
instance-id [YourInstanceId] --device /dev/sda1
```

After issuing the command, the State should change to "attaching."

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
C:\Users\Administrator>aws ec2 attach-volume --volume-id vol-1f7f4b55 --instance
-id i-526b343b --device /dev/sda1
{
    "AttachTime": "2013-08-30T22:24:32.090Z",
    "InstanceId": "i-526b343b",
    "VolumeId": "vol-1f7f4b55",
    "State": "attaching",
    "Device": "/dev/sda1"
```

66. Type the following command to view the status of the attachment (replacing [YourNewVolumeId] with the id of the new, larger volume):

```
aws ec2 describe-volumes --volume-ids [YourNewVolumeId]
```

67. When the larger volume is successfully attached, the State will change to "in-use." You will also see the new Size reflected: 120GB.

```
        "VolumeType": "standard",
        "VolumeId": "vol-1f7f4b55",
        "State": "in-use",
        "SnapshotId": "snap-1632d819",
        "CreateTime": "2013-08-30T21:53:47.000Z",
        "Size": 120
```

68. Type the following command to start the storage instance using the new, larger volume (replacing [YourInstanceId] with the id of the storage instance):

```
aws ec2 start-instances --instance-id [YourInstanceId]
```

After issuing the command, the State will change to "pending."

```
C:\Users\Administrator>aws ec2 start-instances --instance-id i-526b343b
{
    "StartingInstances": [
        {
            "InstanceId": "i-526b343b",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
```

69. Type the following command to view the State of the instance (replacing [YourInstanceId] with the id of the storage instance):

```
aws ec2 describe-instances --instance-ids [YourInstanceId]
```

The instance is started when the State changes to "running."

```
"Instances": [
    {
        "Monitoring": {
            "State": "disabled"
        },
        "PublicDnsName": null,
        "Platform": "windows",
        "State": {
            "Code": 16,
            "Name": "running"
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

70. Type the following command to view the status of the new volume (replacing [YourNewVolumeId] with the id of the new, larger volume):

```
aws ec2 describe-volumes --volume-ids [YourNewVolumeId]
```

The new, 120GB volume should be "attached" and the State should be "in-use."

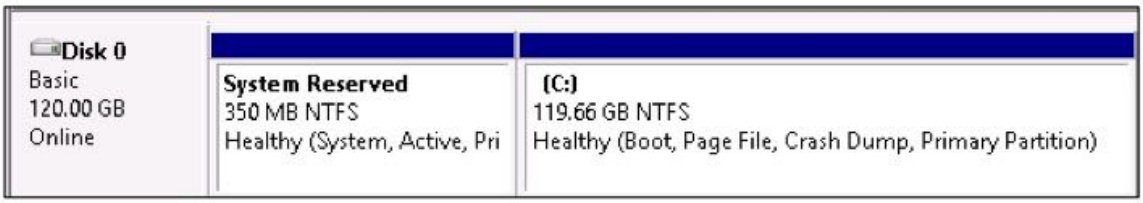

```
"Volumes": [
    {
        "AvailabilityZone": "us-east-1c",
        "Attachments": [
            {
                "AttachTime": "2013-08-30T22:24:32.000Z",
                "InstanceId": "i-526b343b",
                "VolumeId": "vol-1f7f4b55",
                "State": "attached",
                "DeleteOnTermination": false,
                "Device": "/dev/sda1"
            }
        ],
        "VolumeType": "standard",
        "VolumeId": "vol-1f7f4b55",
        "State": "in-use",
        "SnapshotId": "snap-1632d819",
        "CreateTime": "2013-08-30T21:53:47.000Z",
        "Size": 120
```


Connect to the Storage Server instance and view your disk storage

In this section of the lab, you connect to your Storage Server instance using the remote desktop client from within your bastion instance.

71. From your bastion instance, log in to the storage instance using your RDP client. Use the same password that you used to connect to the bastion instance. Refer to previous steps if you need assistance.
    Note: It may take a moment (up to 10 minutes) for your login to complete due to the User Profile Service.

72. Click the **Server Manager** link in the taskbar.
73. From the menu options, choose **Tools > Computer Management**.
74. Expand **Storage** and click **Disk Management**.
75. Notice that your disk manager shows 90 GB of unallocated space.
76. Right click the **C:** drive block and select **Extend Volume**.
77. Click **Next**. Notice that by default, all of the available space remaining on the volume is selected.
78. Click **Next** and then click **Finish**. Notice the disk is now extended.



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: This can also be done at the command-line using diskpart. For more information on diskpart, see: http://technet.microsoft.com/en-us/library/cc776741.aspx.

79. Minimize the remote desktop client session with the storage instance (you use it later in this lab).
80. Using the command-line interface on the bastion instance, type the following commands to delete the old 35GB volume and the snapshot (replacing [YourOldVolumeId] with the id of the older, smaller volume and replacing [YourSnapshot] with the id of the snapshot you created):

```
aws ec2 delete-volume --volume-id [YourOldVolumeId]
aws ec2 delete-snapshot --snapshot-id [YourSnapshot]
```

The output of both commands should be: "return": "true."

Summary: Part 2

You have successfully increased the disk space available to an EC2 instance with minimal downtime. You can script these types of tasks rather than issuing commands using the command-line interface. EC2 provides powerful APIs and tooling as an alternative to traditional infrastructure management. Next we will create a provisioned IOPS volume and attach it.

Part 3: Creating Provisioned IOPS and Striped Volumes

Part 3 of this lab requires the instance ID of your Storage Server instance. If you did not record if previously, please refer back to the command set to retrieve it.

81. Follow these steps to attach a high speed Provisioned IOPS (PIOPS) volume to your storage instance.
82. In the CLI on your bastion instance, type the following command to create a new PIOPS volume (replacing [YourAZ] with your Availability Zone)

```
aws ec2 create-volume --availability-zone [YourAZ] --
size 200 --volume-type io1 --iops 2000
```

Note: The ratio for disk size (in GB) to IOPS cannot exceed 10:1.

93. Record the volume id.
94. Type the following command to verify that the volume state has changed from "creating" to "available" (replacing [YourVolumeId] with the volume id of the PIOPS volume):

```
aws ec2 describe-volumes --volume-id [YourVolumeId]
```

95. Type the following command to attach the new volume to your storage instance (replacing [YourVolumeId] with the id of the PIOPS volume and replacing [YourInstanceId] with the id of the storage instance):

```
aws ec2 attach-volume --volume-id [YourVolumeId] -
-instance-id [YourInstanceId] --device /dev/xvdb
```

96. Next you create additional volumes and stripe them. Type the following commands (replacing [YourAZ] with your Availability Zone):

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
aws ec2 create-volume --availability-zone [YourAZ] --
size 100
aws ec2 create-volume --availability-zone [YourAZ] --
size 100
```

Note: You can stripe across more than two volumes – this is presented as an example.

Also note: This is an alternate way to get increased IOPS; however, it is not a fault tolerant solution. It may be useful for temporary processing where failure can be handled (such as processing data jobs where the job can start over if data is lost). You can also create the additional volumes as PIOPS volumes and combine them to exceed the current AWS PIOPS limit.

97. Type the following command to view the state of the two new volumes (replacing [YourVolumeId] with the volume id of each new volume). Verify that they are both "available" before continuing:

```
aws ec2 describe-volumes --volume-ids [YourVolumeId]
[YourVolumeId]
```

98. Attach the new volumes to your storage instance by typing the following commands (replacing [YourVolumeId] with the volume id of each new volume and replacing [YourInstanceId] with the id of the storage instance):

```
aws ec2 attach-volume --volume-id [YourVolumeId] --
instance-id [YourInstanceId] --device /dev/xvdc
aws ec2 attach-volume --volume-id [YourVolumeId] --
instance-id [YourInstanceId] --device /dev/xvdd
```

99. In your remote desktop session with the storage instance, open the **Disk Management** utility (refer to previous steps if you need assistance). You should see each of the three volumes you created (each will be unallocated and Offline).
100. Right-click the 200GB disk and choose **Online**.
101. Repeat the previous step to bring the other 2 disks online.
102. Right-click the 200GB disk again and choose **Initialize Disk**.
103. In the **Initialize Disk** dialog, leave all 3 drives selected and click **OK**.
104. Right-click the unallocated space for the 200GB disk and choose **New Simple Volume**.
105. At the **Welcome** panel, click **Next**.
106. At the **Specify Volume Size** panel, accept the default value and click **Next**.
107. Accept the default drive letter and click **Next**.
108. At the **Format Partition** panel, accept the default values to format the partition as **NTFS** and click **Next**.
109. Click **Finish** then wait for the drive to be formatted and for a drive letter to be assigned.
110. Right-click one of the unallocated, 100GB volumes and choose **New Striped Volume**.
111. At the **Welcome** panel, click **Next**.
112. At the **Select Disks** panel, add the second 100GB volume to the **Selected** window and click **Next**.
113. Accept the default drive letter for the volumes and click **Next**.
114. Accept the default format (NTFS) and click **Next**.
115. Click **Finish**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

116. When you receive the message "The operation you selected will convert the selected basic disk(s) to dynamic disk(s)," click **Yes**.
117. The Disk Management utility should be updated with your new disk properties:

Summary: Part 3

In this lab you created two types of high-speed drives. One was a PIOPS drive. The other was comprised of two striped volumes.

Conclusion

Having completed this lab you should be able to:

- Use a command-line interface to configure an EC2 Windows instance
- Connect to EC2 instances
- Create EBS volumes
- Create and instantiate snapshots
- Stripe volumes
- Create EBS provisioned IOPS volumes
- Use the AWS command-line tool

Appendix A

EC2 Calls from PowerShell using Cmdlets

1. Download and install Windows PowerShell, which is part of the Windows Management Framework available here: http://www.microsoft.com/en-us/download/details.aspx?id=34595
    a. Select the .msu file appropriate for your version of Windows. If you are running a 64-bit version of Windows, you can get Windows6.1-KB2506143-x64.msu.
2. Download and install AWS Tools for Windows PowerShell available here: http://aws.amazon.com/powershell/
3. See this page for details on installing and setting up AWS Tools for Windows PowerShell: http://docs.aws.amazon.com/powershell/latest/userguide/pstools-getting-set-up.html
4. For setting credentials in your PowerShell session, see the instructions here: http://docs.aws.amazon.com/powershell/latest/userguide/specifying-your-aws-credentials.html
    a. Initialize-AWSDefaults -AccessKey AKIAIOSFODNN7EXAMPLE -SecretKey wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEK -Region *xy-region-1*
5. Smoke test that the AWS Tools for Windows PowerShell are working correctly:
    a. When you finish the steps shown previously, you should see AWSPowerShell as one of the available services by running this command:
    Get-Module -ListAvailable
    b. You can list all of the available cmdlets available in the AWSPowerShell module by running this command:
    Get-Command -Module AWSPowerShell

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

c. As you can see, the list is quite long. You can filter for only EC2 related cmdlets by running:
Get-Command -Module AWSPowerShell | findstr EC2

6. Get a list of all EC2 machines that you own by running the following PowerShell command:
Get-EC2Image -Owner self
7. You can describe a particular attribute of a particular EC2 instance by running the following command:
Get-EC2InstanceAttribute YourInstanceIdHere AttributeNameHere
Example: Get-EC2InstanceAttribute i-aaaaaa instanceType
Example: Get-EC2InstanceAttribute i-aaaaaa rootDeviceName

A complete list of available attributes is listed here:
http://docs.aws.amazon.com/AWSEC2/latest/APIReference/ApiReference-query-DescribeInstanceAttribute.html

EC2 Calls from PowerShell using AWSCLI

1. If you were running AWS CLI from PowerShell, you may want to run a ps1 script to initialize your environment settings for your console session. We might name the script something like InitAWSCLI.ps1 and put it in the C:\aws folder A script to do that might look something like this:

```
$env:AWS_CONFIG_FILE = "C:\aws\AWSCLIConfig.txt"
Write-Output "AWS config file location set"
Write-Output "Welcome to the unified AWS command line
interface!"
Write-Output `n
```

2. Next we have to ensure that Powershell will run our scripts. Right click the Powershell shortcut and click Run as administrator, then run the following commands:

```
Set-ExecutionPolicy RemoteSigned
[ENTER]
exit
```

3. Run Powershell by clicking the shortcut, and run the following commands:

```
cd C:\aws
.\InitAWSCLI.ps1
cd C:\aws
.\InitAWSCLI.ps1
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

End Your Lab

118. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

119. Close any active SSH client sessions or remote desktop sessions.
120. Click the **End Lab** button on the *qwikLAB*™ lab details page.

121. When prompted for confirmation, click **OK**.
122. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
Tagging
Windows Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

Tags are an essential part of keeping a large AWS footprint organized. Tags can be used to sort AWS resources, and can also be used to search for particular resources. Although rather simple in and of themselves, tags are one of the fundamental building blocks of effectively managing a dynamic infrastructure.
In this lab you will:

- Sort instances by tag using the AWS Management Console
- Search for tags using the AWS Management Console
- Search for tags using the Amazon EC2 CLI
- Create custom tags using the AWS Management Console
- Create custom tags using the Amazon EC2 CLI

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with using RDP to access Windows servers.

Topics Covered

This lab will focus entirely tagging in the context of EC2 instances; tags for other AWS resources utilize the same concepts, but will not be introduced in this lab.

Connect to the Bastion Host

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair

This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this guide called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group

It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

1. Retrieve your current public IP address by navigating your browser to http://ip.appspot.com
2. Open the **AWS Management Console**.
3. Select the **VPC** service.
4. From the menu on the left side, click **Security Groups**.
5. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion Host"**.
6. On the bottom pane, click the **Inbound** tab.
7. From the **Create a new rule dropdown**, select RDP.
8. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
9. Click the **Add rule** button.
10. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host

11. Open the **AWS Management console**.
12. Select the **EC2** service.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

13. From the menu on the left side, click **Instances**.
14. **Select the BastionHost instance** by clicking on the box next to its name.
15. The Public IP is displayed on the bottom pane. Please write it down.

Retrieve the bastion instance password
Before connecting to the bastion using RDP, you must retrieve the instance's password for the Windows Administrator account.
16. In your browser, switch to the **qwikLAB** tab.
17. In the **Lab Connection** portion of the page, copy the **Password** in the **RDP** section – you may also paste it into a text file for later retrieval.

Note: This is the password you use to log into the bastion instance using the remote desktop client.
18. If the computer you are using to complete this lab is Windows-based, proceed to the next section. If you are using an OS X or Linux-based machine, proceed to the section appropriate to your operating system.

Connecting to the bastion instance (Windows)
Important: Proceed to one of the later sections entitled "Connect to the bastion instance (OS X)" or "Connect to the bastion instance (Linux)" if you are using a Linux-based laptop to perform this exercise. This section is *only* for Microsoft Windows users. There is no need for you to perform the steps in all sections. Only complete the section that matches your operating system.
19. On your local computer click **Start > Run**, type **MSTSC** and click **OK** to start the local RDP client.
20. Type or paste the Elastic IP address of the bastion instance in the **Computer** field.
21. Click **Show Options**, and type **\Administrator** in the **User name** field.

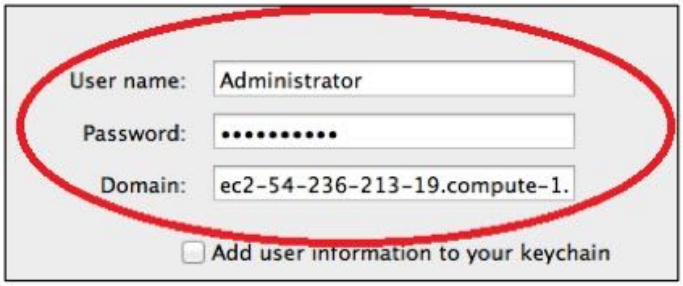
Note: Specifying the user name as "\Administrator" (with a leading backslash) differentiates the bastion Administrator account from the Administrator account on your local computer.
22. Click **Connect**.
23. When prompted, type the RDP **Password** you noted then click **OK**.
24. Click **Yes** if you see a certificate verification message similar to "the identity of the remote computer cannot be verified:"

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

25. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (OS X)
Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." This section is for OS X users *only*. If you are using a different Linux-based operating system, please proceed to the section entitled "Connect to the bastion instance (Linux)." There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

26. Open the Remote Desktop Connection for Mac application.
27. Type the bastion instance's IP address you noted in the **Computer** field and click **Connect**.
28. When prompted, in the **User name** field, type Administrator and then type the **Password** you noted from the qwikLAB page. The Domain should be automatically set to the EC2 Instance DNS name, and you can ignore it.



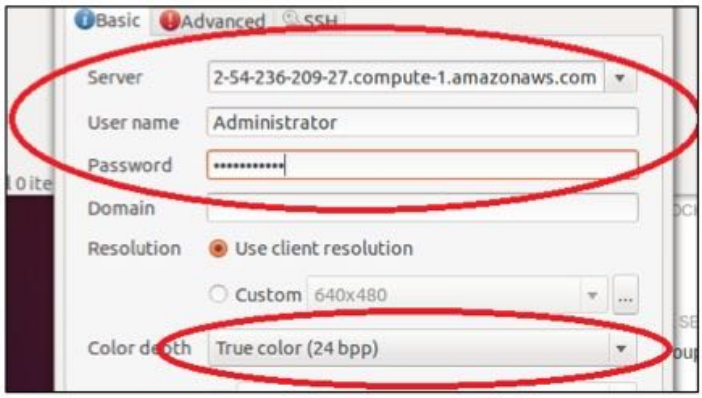
29. Click **OK**.
30. Click **Connect** if you see a verification message similar to "the server name is incorrect."
31. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (Linux)
Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." Only complete this section if you are using a non-

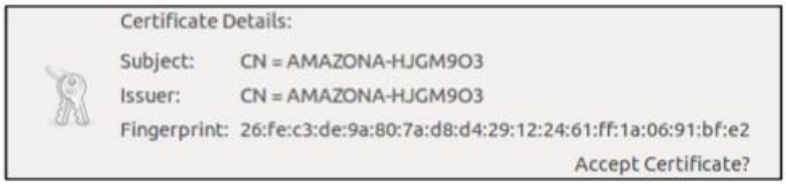
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Macintosh, Linux-based laptop to perform this exercise. This section is *only* for Linux users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

32. Open the Remmina Remote Desktop Client.
33. Type the bastion instance's IP address you noted in the **Server** field.
34. In the **User name** field type **Administrator**, and in the **Password** field, type the password from the qwikLAB page.
35. Optionally, choose a **Color depth** that your bandwidth supports (in this example 'True color (24 bpp)') for a nicer remote desktop environment.



36. Click **Connect**.
37. Click **OK** when prompted to accept the remote certificate.

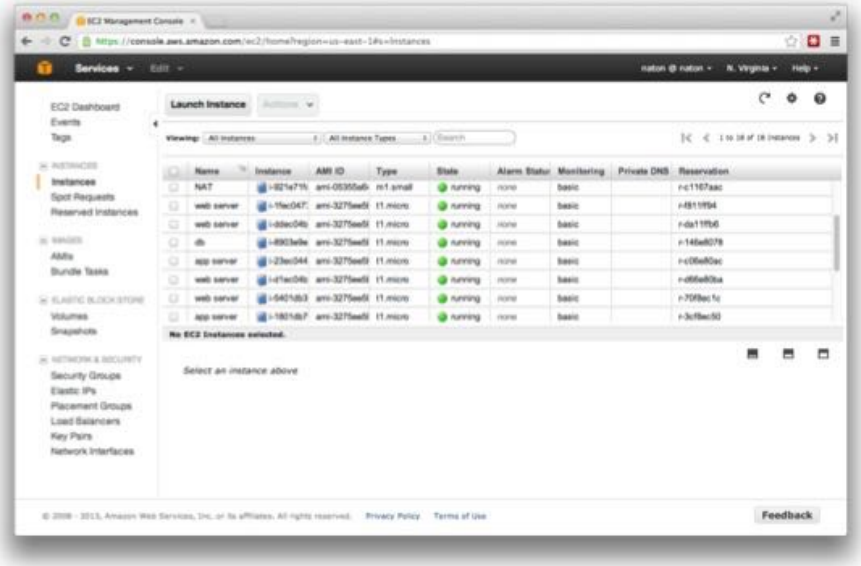


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

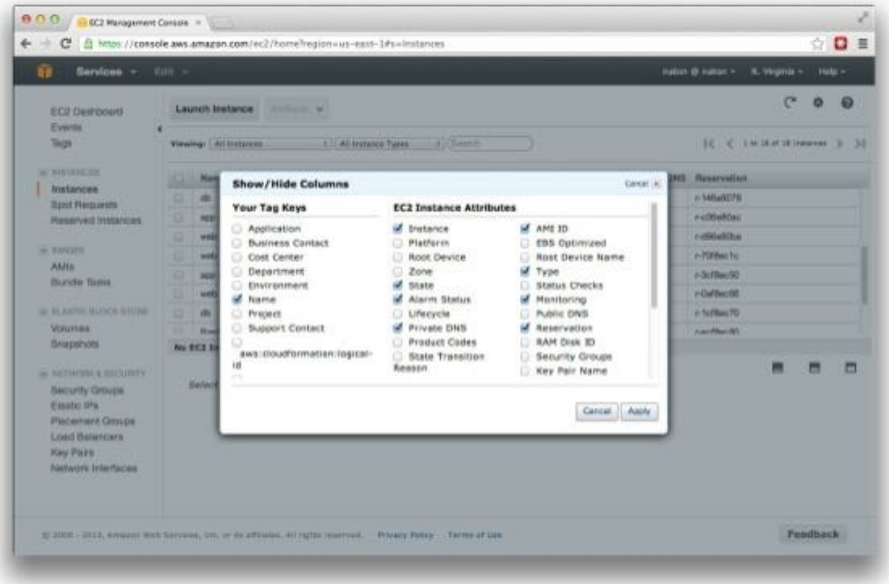
Working with tags in the console

Sorting and searching by tags

1. Log into the AWS Management Console and navigate to **EC2->Instances->Instances** (ensure that the correct region is selected). There should be at least 10 running instances visible.



2. Click on the Show/Hide icon.



3. Under the heading "Your Tag Keys", select the checkboxes for "Application", "Name", and "Environment". Click "Apply". Click on the Refresh icon next to the Show/Hide icon.

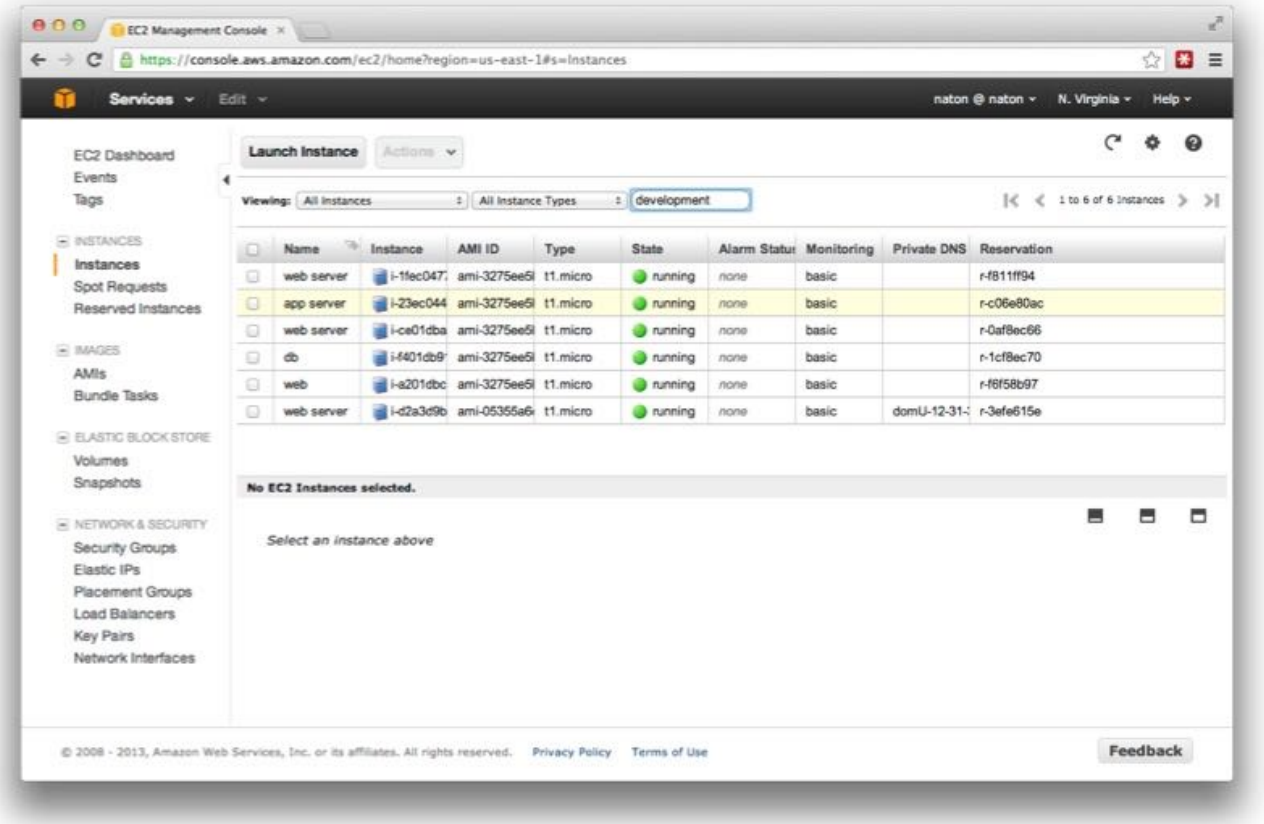
Sort

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

4. Click on the "Application" column to sort all instance by application. Note how many servers are dedicated to each application.

Search

5. In the search bar, type "development" to only show instances with "development" in their tag values.

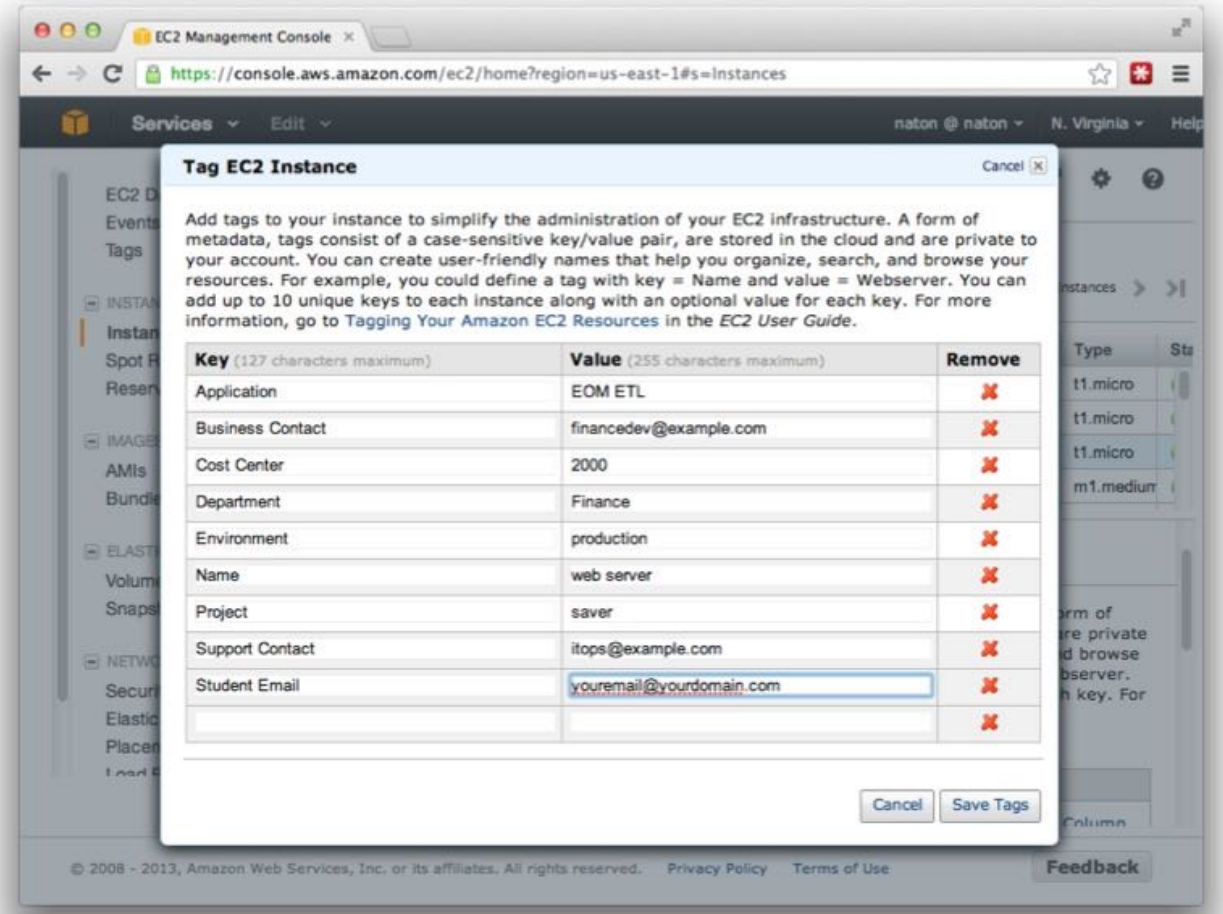


Get / change instance tags

6. Click on the checkbox for any one (but only one) of the instances. In the bottom half of the screen, navigate to the "Tags" tab and scroll through the tags. Notice the tags at the bottom that start with "aws:" – these are immutable and cannot be deleted.

7. Take note of the instance id then, click on the "Add/Edit Tags" button, and add a tag with key "Student Email", and your email address as the value. Click on "Save Tags" to apply.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



Working with the tagging CLI

8. Using your preferred RDP client, log into the BASTION Windows host (see Appendix A and the appropriate "Connect to the Windows Bastion Server" segment).

[Note about commands: if you choose to run these commands in PowerShell, you will need to escape all quotes like so: command "`"quoted text`"" – for ease of copy/paste, all commands are shown as they would be entered in a cmd window]

9. Try running the command to show all tags:

```
aws ec2 describe-tags
```

This will show every tag for all EC2 resources. This generally isn't very useful, but could be used as the basis for a script to gather information, search, or take other action.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

10. Search explicitly for the tags pertaining to the instance that was just edited in the AWS Management Console:

```
aws ec2 describe-tags --filters "Name=resource-id,Values=i-
01234567"
```

[Note:in all these examples, you will need to replace **i-01234567** with the correct instance id]

11. Alternately, try searching for any tags with the key "Student Email":

```
aws ec2 describe-tags --filters "Name=key,Values=Student Email"
```

Note that this only comes up with the tag that was added from the console.

12. You can add or modify tags using the **aws ec2 create-tags** command. Try changing the number of the Cost Center in the instance you just modified:

```
aws ec2 create-tags --resources i-01234567 --tags "Key=Cost
Center,Value=2001"
```

13. Refresh the console to reflect the change.
14. Finally, delete the custom tag created previously:

```
aws ec2 delete-tags --resources i-01234567 --tags "Key=Student
Email"
```

Bonus activities (solutions on next page):

- Find all instances associated with Cost Center 1000
- Change all tags of "Cost Center=1000" to "Cost Center=1001"

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Bonus answers:

15. Find all instances associated with Cost Center 1000: there's more than one way to do this – here's one way in PowerShell:

    [Note: replace **<your-region>** with the applicable region]

```
Initialize-AWSDefaults -Region <your-region>
Get-EC2Instance -Filter @{Name="tag:Cost Center";Value="1000"} |
Select-Object -ExpandProperty RunningInstance | Select InstanceId
```

16. We can build on this to solve the next one: Change all tags of "Cost Center=1000" to "Cost Center=1001":
    [Note: replace InstanceID with the instance id you gathered previously]

```
New-EC2Tag -Tag @{Key='Cost Center'; Value='1001'} -ResourceId
InstanceId
```

Conclusion

Congratulations! You now have successfully:

- Used the console to sort and filter instances by tags
- Used the console to manage and change tags
- Used the CLI to list and search by tags
- Used the CLI to edit and delete tags

End Your Lab

17. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

18. Close any active SSH client sessions or remote desktop sessions.
19. Click the **End Lab** button on the *qwikLAB*™ lab details page.

20. When prompted for confirmation, click **OK**.
21. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS

Monitoring Lab
Windows Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

In this lab you are going to develop a monitoring solution using Amazon CloudWatch. You are going to learn how to interact with several AWS services using the CLI, PowerShell, and the web console, and how to integrate with third party monitoring solutions.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with the Windows Powershell and using Remote Desktop to access Windows servers.

Topics Covered

This lab will demonstrate how to:

- Write and read monitoring statistics from CloudWatch
- Create alerts and notifications
- Integrate with 3rd party monitoring solutions (enterprise or cloud based)

Interacting with Amazon CloudWatch

Overview

Amazon CloudWatch provides monitoring for AWS cloud resources and the applications customers run on AWS. Developers and system administrators can use it to collect and track metrics, gain insight, and react immediately to keep their applications and businesses running smoothly. Amazon CloudWatch monitors AWS resources such as Amazon EC2 and Amazon RDS Database instances, and can also monitor custom metrics generated by a customer's applications and services. With Amazon CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health. In this section, you are going to interact with CloudWatch writing and reading monitoring statistics.

Writing and Reading Monitoring Statistics

Amazon CloudWatch automatically collects statistics from your AWS resources, such as CPU utilization from EC2 instances, latency from ELBs, and other metrics from several services. Moreover, you can send arbitrary statistics from your instance or application to CloudWatch, thus improving your monitoring solution.

The first section of this will send and retrieve a custom metric to and from CloudWatch.

Create an EC2 IAM Role

The monitoring client will need authorization to write statistics to CloudWatch. Using IAM Roles you can add the proper authorization policy without setting explicit credentials, being more convenient and secure. To create an IAM Role with write access to CloudWatch, follow these steps:

1. Navigate to the IAM Management Console
2. Select **Roles** in the navigation menu
3. Start the **Create New Role** tool
4. Name your new role (e.g. `MonitoringClient`) and **Continue**
5. In the **AWS Service Role** list, select **Amazon EC2**, indicating that this role will be used by EC2 instances
6. In the **Policy Template** list, select the **CloudWatch Full Access template, granting read and write access to CloudWatch.**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

7. **Review the policy document. It allows the API actions required to read and write to CloudWatch.**
8. **Continue and then Create Role.**

In the end of this step, you should see the new role in the IAM Management Console.

Launch an EC2 Windows 2013 Instance

9. Navigate to the EC2 Management Console
10. Click **Launch Instance**
11. Select the **Microsoft Windows Server 2012 Base** AMI
12. Select the **m1.medium** instance type found under the **General Purpose** family.
13. Proceed with **next: Configure Instance Details**
14. Accept default settings for **Network** (default VPC) and for **Subnet** (No Preference)
15. Select the **IAM Role** created in the previous section
16. **Review and Launch** the instance, after selecting your key pair (qwikLABS-*.pem). You can accept default security group and tagging settings.

In the end of this section you should see a new instance with the selected role in the **IAM Role** property. Wait until the instance is ready to proceed. This instanced will be referenced as "monitoring instance" in the following instructions

Write the new metric data

In this section you are going to write arbitrary monitoring data from your EC2 instance using the AWS Command Line Interface for Windows.

17. Access the monitoring instance using Remote Desktop. It may take several minutes until the Administrator password is available.
18. Download and install AWS Command Line Interface for Windows (64-bit)

https://s3.amazonaws.com/aws-cli/AWSCLI64.msi

19. Open a Powershell Command Line in the monitoring instance.
20. Set the default AWS Region for CLI commands using an environment variable

```
[Environment]::SetEnvironmentVariable("AWS_DEFAULT_REGION", "<your-aws-region-id>", "User")
```

21. Open a new Powershell prompt and send a new metric data to CloudWatch, with any meaning. As an example, you can use "attention level" as the metric name. You can have any meaning and range of values to the metrics you define:

```
aws cloudwatch put-metric-data `

  --namespace Student `

  --metric-name AttentionLevel `

  --value 8 `

  --debug
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

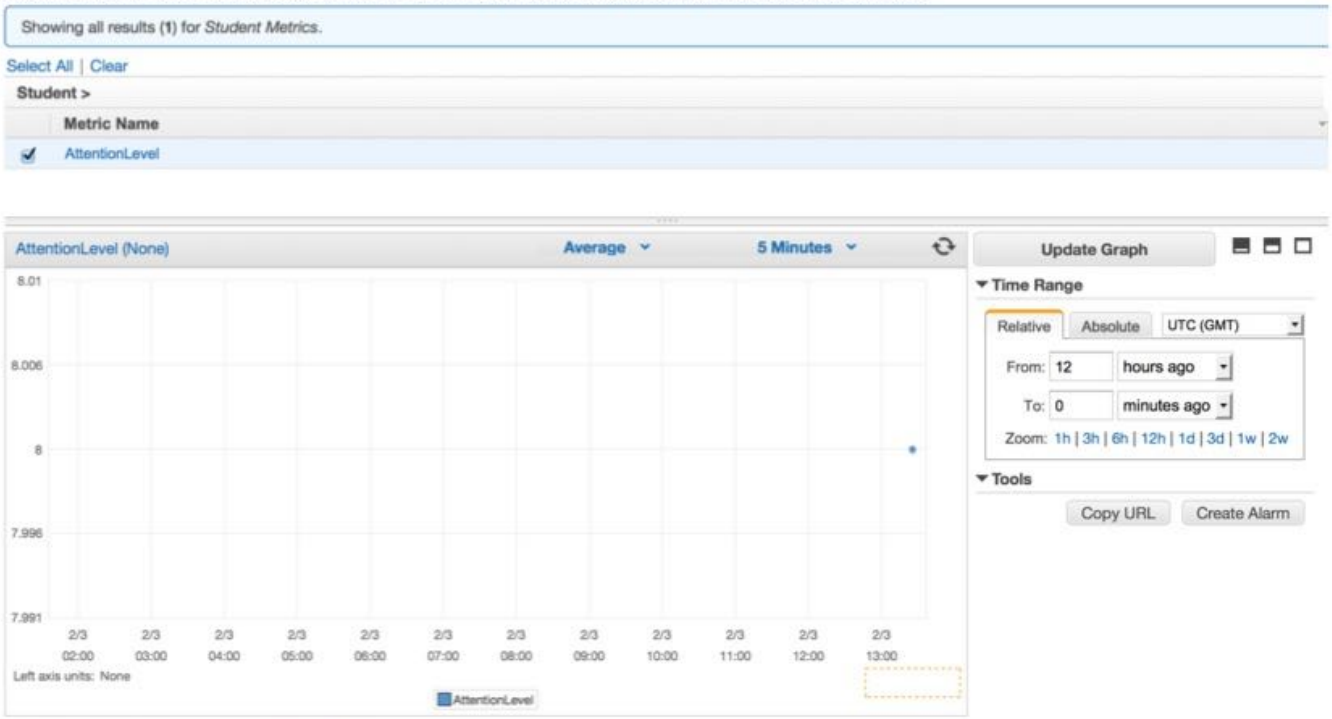
You should see a successful response (*HTTP Response Code: 200*) in the debug output.

Visualize the new metric

In this section you are going to review your monitoring data using the CloudWatch Management Console.

22. Open the CloudWatch Management Console
23. Select the *Student* namespace under **Custom Metrics** in the navigation menu and then the *AttentionLevel* metric

At the end of this section you should see a graph displaying your custom metric.



Read statistics using the CLI

In this section you are going to retrieve your monitoring data from CloudWatch using the AWS Command Line Interface for Windows.

24. Get the minimum attention level statistic from the shell using the CLI:

```
$yesterday=[DateTime]::Today.AddDays(-1).ToString("d")

$tomorrow=[DateTime]::Today.AddDays(1).ToString("d")

aws cloudwatch get-metric-statistics `

 --metric-name "AttentionLevel" `
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
 --namespace="Student" `


 --start-time=$yesterday `

 --end-time=$tomorrow `

 --period=300 `
 --statistics="Minimum"
```

At the end of this section you should see your monitoring statistics in the console.

Alarms and Notifications

In this section you are going to create an alarm to notify you when the metric value drops below a given threshold:

25. Return to the browser widow displaying your custom metric graph
26. Start the **Create Alarm** tool
27. Configure your **Alarm Threshold**

    Name: Attention too low
    Description: Alert when attention level drops below 6
    Whenever: AttentionLevel
    is: < 6
    for: 1 consecutive period
28. Set the **Notification Action** and **Create New Topic**

    Whenever this alarm: State is ALARM
    Follow the create topic link
    Send Notification To: SystemOperator
    Email list: [your@email.com]
29. **Create Alarm**
30. Confirm the alarm subscription request sent to your email
31. Generate a data point that triggers the alarm

```
aws cloudwatch put-metric-data `

  --namespace Student `

  --metric-name AttentionLevel `

  --value 4 `
 --debug
```

32. Navigate to your alarm and see that it has changed state from OK to ALARM
33. Wait for the email notification.

At the end of this section you should have received a low memory notification.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Integrating with Other Monitoring Services

Your monitoring solution may include 3rd party services, such as on-premise monitoring software or cloud monitoring services. In those cases, you have a couple options to register and unregister monitoring clients (your instances) as they are started and stopped.
To demonstrate the three scenarios presented in the course, you are going to start a "mock" monitoring server and then see how the different approaches work. The mock server is a simple registry of clients with an HTTP API, accepting requests with the following methods:

- GET: Lists the registered instance ids and registration time
- PUT or POST: Register the instance specified by the **instanceId** parameter
- DELETE: Clears the registry

Start the Monitoring Server

34. Access the monitoring instance using Remote Desktop
35. Disable the Windows Firewall

```
netsh advfirewall set allprofiles state off
```

36. Download and install Python 2.7 from https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-monitoring-windows-v1.1/static/python-2.7.6.amd64.msi
37. Download and install SetupTools from https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-monitoring-windows-v1.1/static/setuptools-2.1.win-amd64-py2.7.exe
38. Download and install PIP from https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-monitoring-windows-v1.1/static/pip-1.5.2.win-amd64-py2.7.exe
39. Download and start the mock monitoring service. If you changed the default paths during the installers above, please adjust the paths in the following commands:

```
c:\Python27\Scripts\pip install Flask

$wc = New-Object System.Net.WebClient

$url = "https://us-west-2-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-
monitoring-windows-v1.1/static/mon-server.py"

$wc.DownloadFile($url,"c:\\mon-server.py")

c:\Python27\python c:\mon-server.py
```

40. Navigate to the Instances panel of the EC2 Management Console
41. Select your instance and, in the properties panel, copy the values for its **Security groups** and **Private IP**
42. Navigate to the Security Groups panel of the EC2 Management Console
43. Select your instance's security group and copy its **Group ID**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

44. In the **Inbound** tab, add a new rule allowing HTTP as protocol and paste the <Group ID> as source
45. **Apply Rule Changes**
46. Start a new Powershell Command Line prompt in the monitoring instance.
47. Issue an HTTP GET request to your server using its Private IP (including the "http://"). In the next sections this value will be referenced as `<monitoring_server_url>`:

```
(New-Object System.Net.WebClient).DownloadString("<monitoring_server_url>")
```

In the end of this section you should see a message stating that no clients are registered.

Client Self-Registration

In this section you are going to start monitoring clients that registers themselves on their first boot:

48. Navigate to the Launch Configurations console
49. Start the **Create Auto Scaling group** tool
50. Proceed with the **Create Launch Configuration** tool
51. Select the **Windows Server 2012 Base** AMI
52. Select the **m1.medium** instance type
53. Proceed with **next: Configure Details**
54. Name your Launch Configuration (e.g. lcMonitoringLab)
55. Open the **Advanced Details** and paste the following **User data** script.

```
<powershell>

$InstanceId=(New-Object
System.Net.WebClient).DownloadString("http://169.254.169.254/latest/meta-
data/instance-id")

Invoke-RestMethod -Uri http://<monitring_server_url> `

 -Method PUT `

 -Body instanceId=$InstanceId `

 -ContentType application/x-www-form-urlencoded



</powershell>
```

This command will register the instance with the mock server using a http PUT request with the instance id as a parameter. In a real scenario you would replace this with the client registration command from the actual monitoring service.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

56. Replace `<monitoring_server_url>` with your mock monitoring server address (including http://)
57. Proceed with **next: Add Storage**, leaving default settings
58. Proceed with **next: Configure Security Group**
59. **Select an existing security group**, choosing the same security group as **step 43**
60. **Review** and **create Launch Configuration**, selecting your keypair
61. Conclude the **Create Lauch Configuration** tool. **Create Auto Scaling Group** tool will start
62. Name your Auto Scaling Group (e.g. asgMonitoringLab)
63. Set your **Group Size** to *start with 2 instances*
64. Select your default VPC as **Network**
65. Select any **Subnet**
66. Proceed to **Configure scaling policies**
67. Select **Keep this group at its initial size**
68. **Review**, **Create your Auto Scaling Group** and **Close.**
69. Wait for the instances to be launched and ready.
70. In your Powershell Command Line session, list the registered instances with a new http GET request:

```
(New-Object System.Net.WebClient).DownloadString("<monitoring_server_url>")
```

In the end of this section you should see two instances registered at different times in the monitoring server.

Full Scan / Config Rewrite

In this section you are going to re-generate the monitoring configuration periodically.

Grant EC2 access to your monitoring server

71. Navigate to the IAM Management Console
72. Select **Roles** in the navigation menu
73. Select the previously created role (e.g. MonitoringClient)
74. Open the **Attach Role Policy** tool
75. Select the **Amazon EC2 Read Only Access** policy template
76. **Apply policy**

Schedule the scan / config rewrite script

77. Access your monitoring instance using Remote Desktop
78. Start a Powershell Command Line prompt in the monitoring instance
79. Allow script execution

```
Set-ExecutionPolicy unrestricted
```

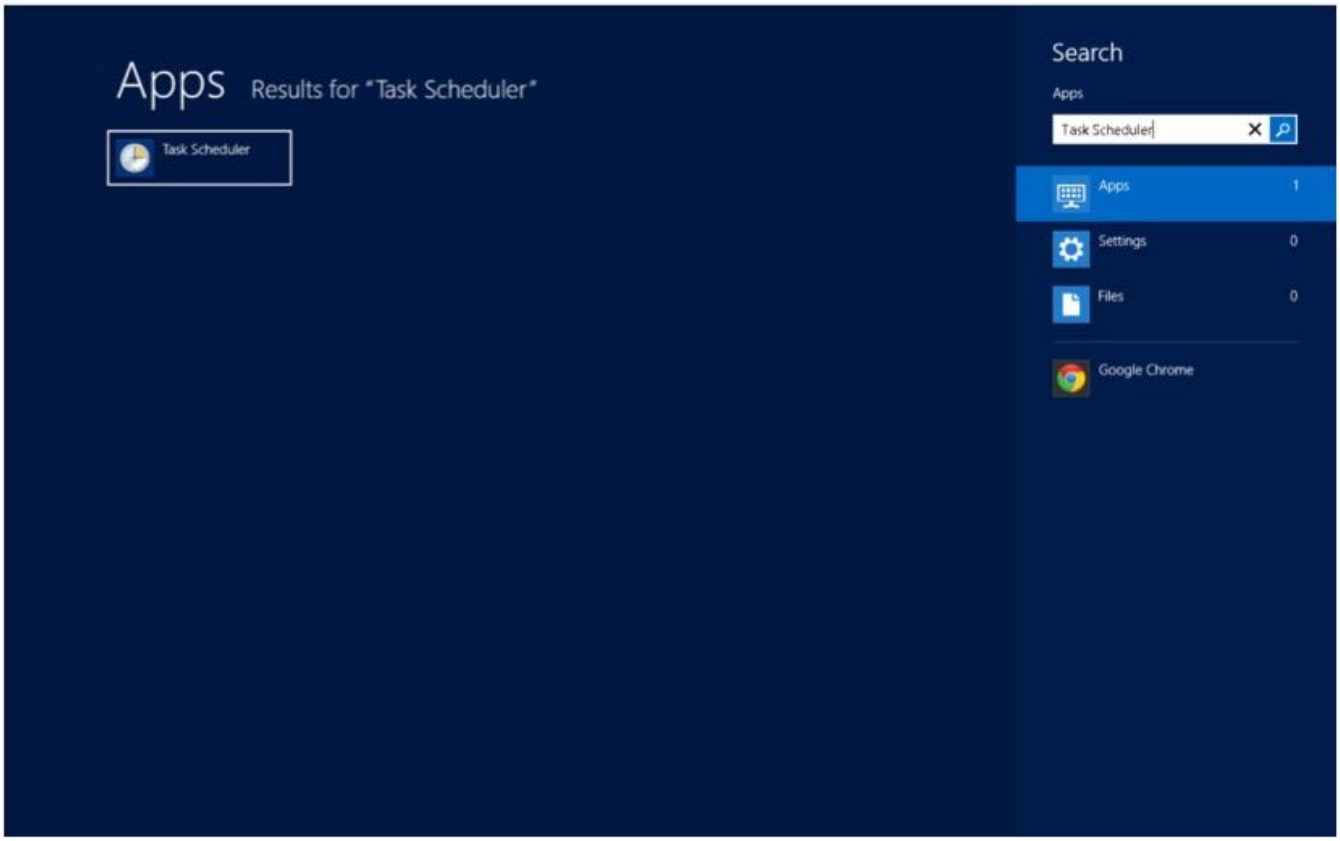
80. Download and run the instance polling script:

```
$wc = New-Object System.Net.WebClient

$url = "https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-
monitoring-windows-v1.1/static/poll-instances.ps1"
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

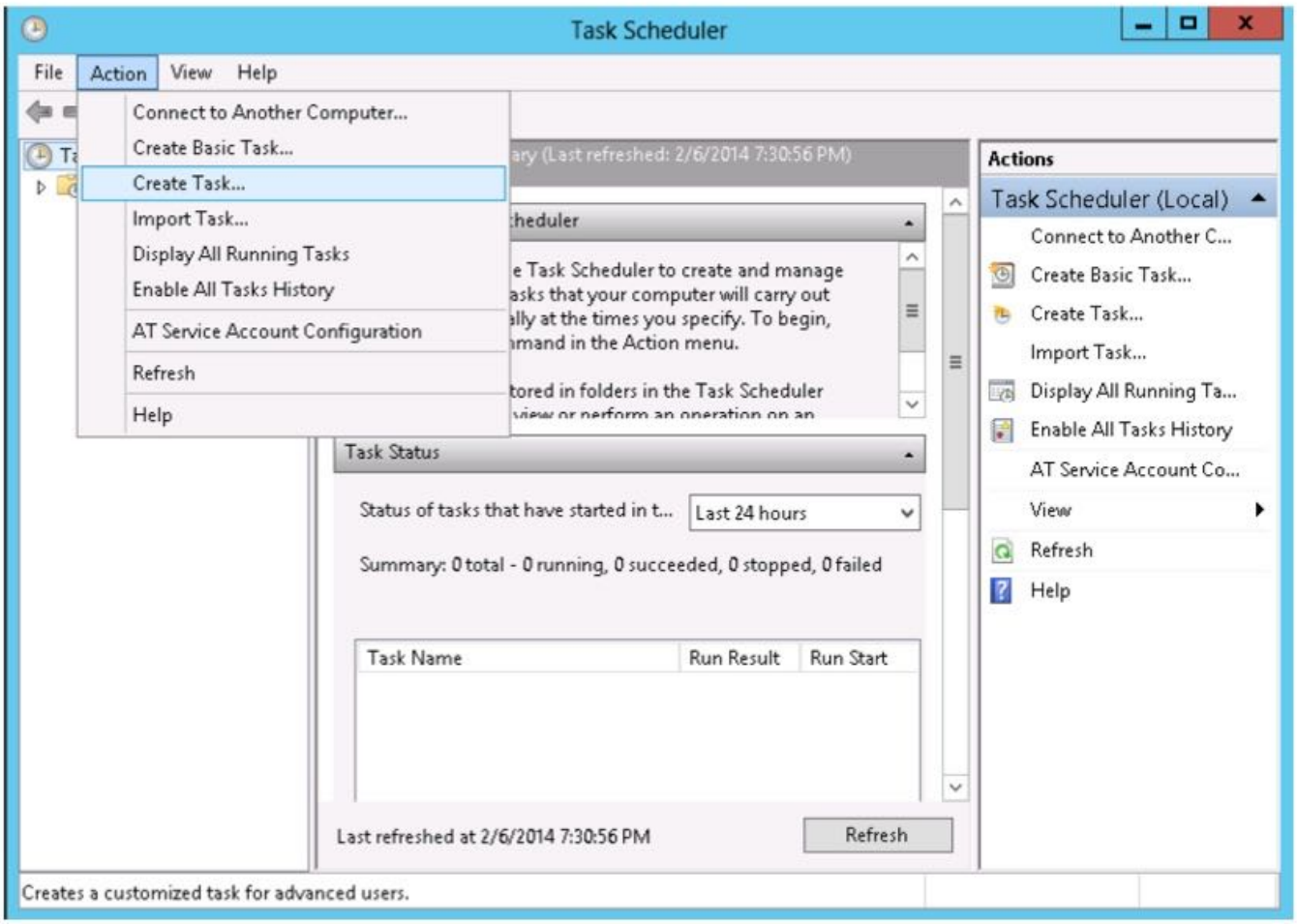
```
$wc.DownloadFile($url,"c:\poll-instances.ps1")

c:\poll-instances.ps1 <monitoring_server_url>
```

81. Schedule the script to run every five minutes using the Windows Task Scheluder. Replace [monitoring_server_url] with the address of your monitoring server, including "http://".
82. Start the Task Scheduler

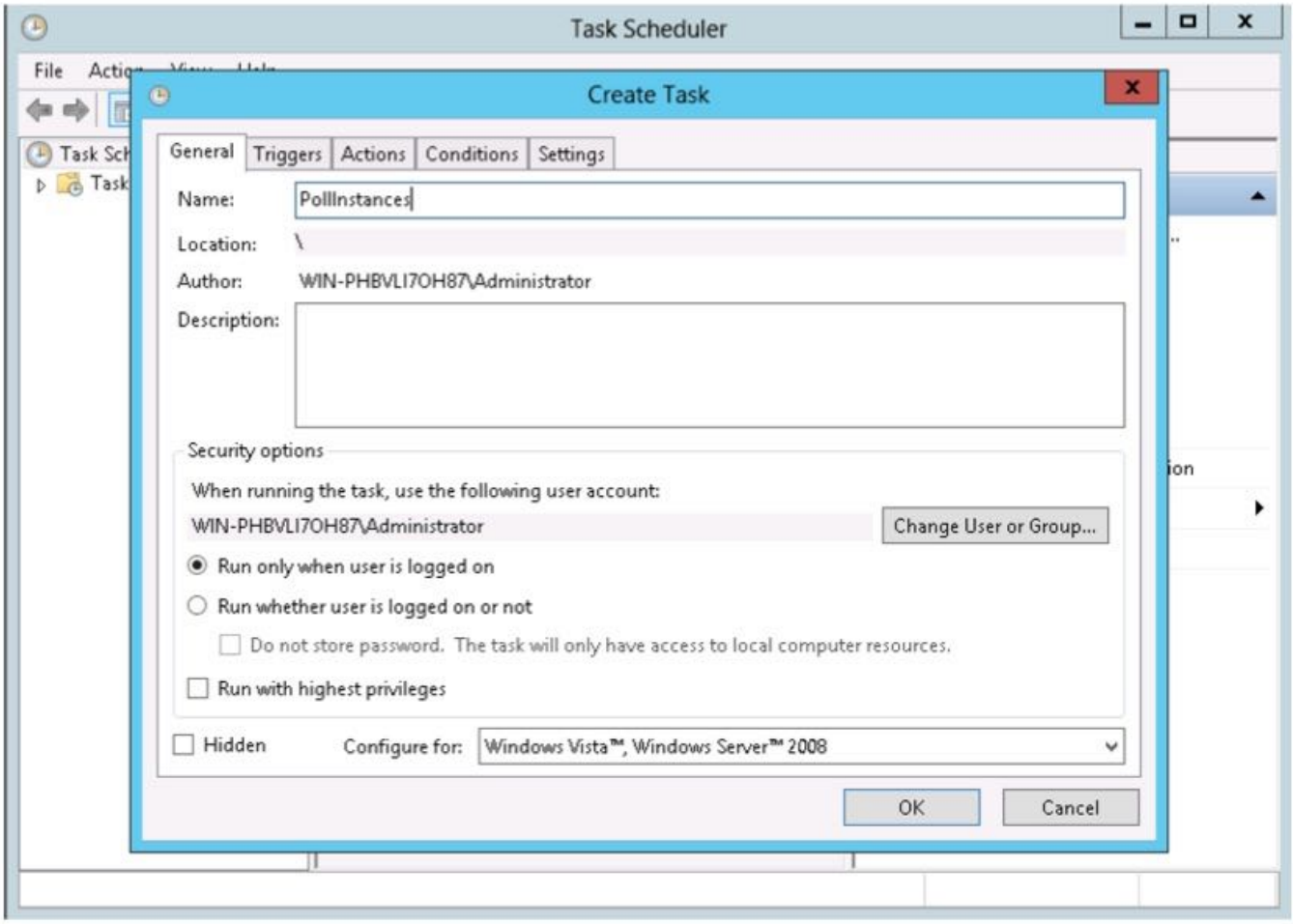


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

83. Create a Task named PollInstances

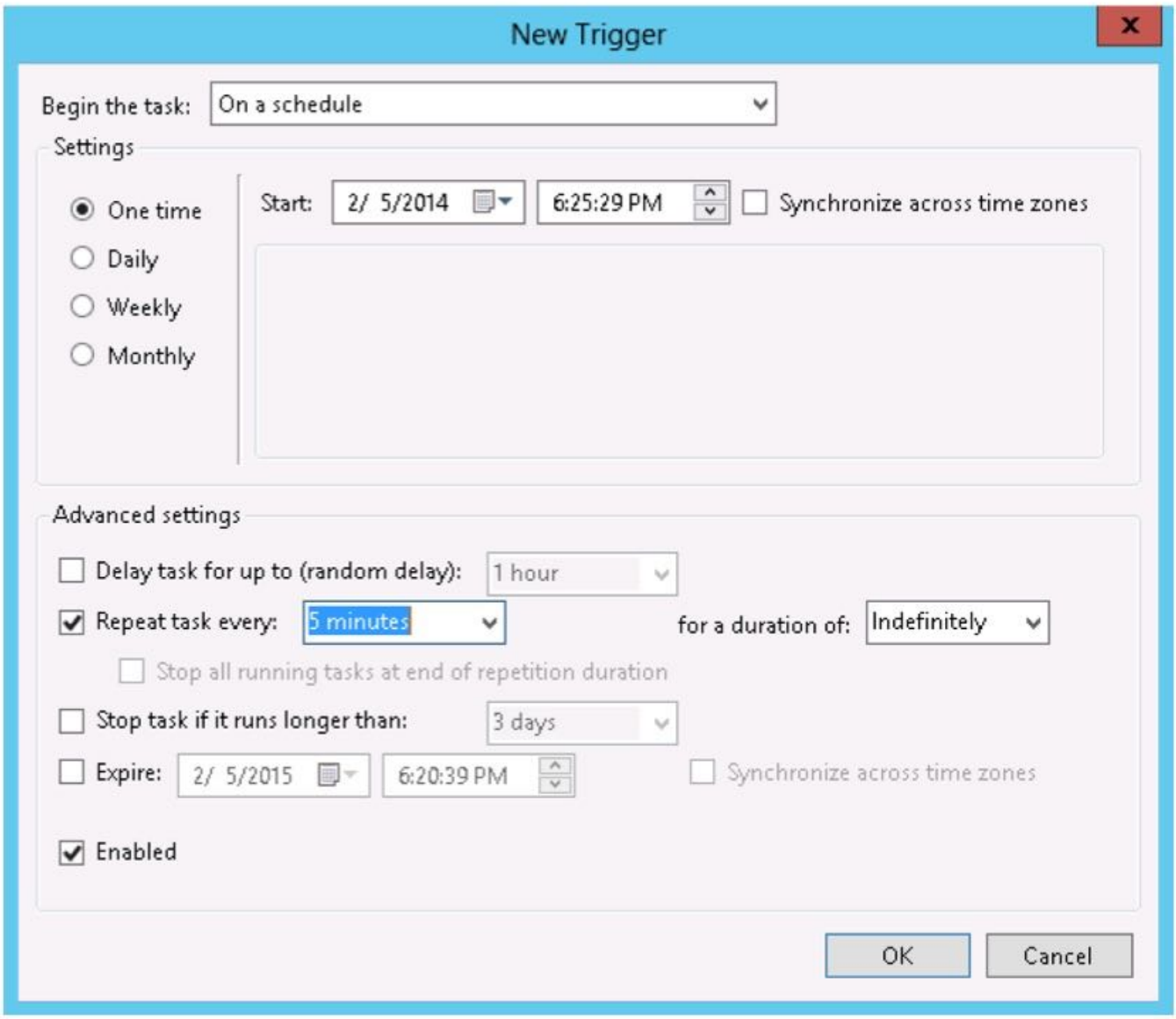


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



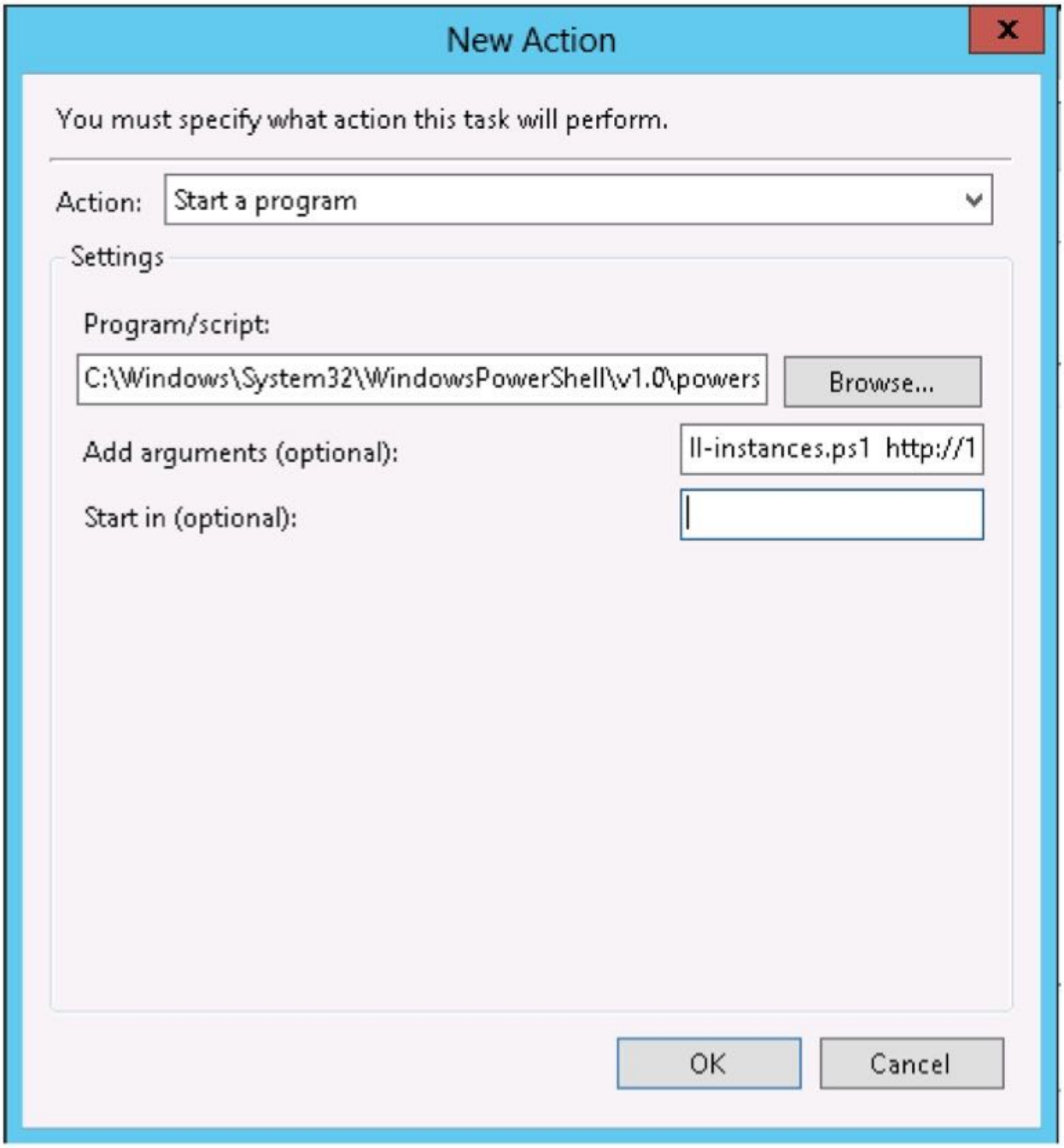
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

84. Add a trigger to execute every five minutes

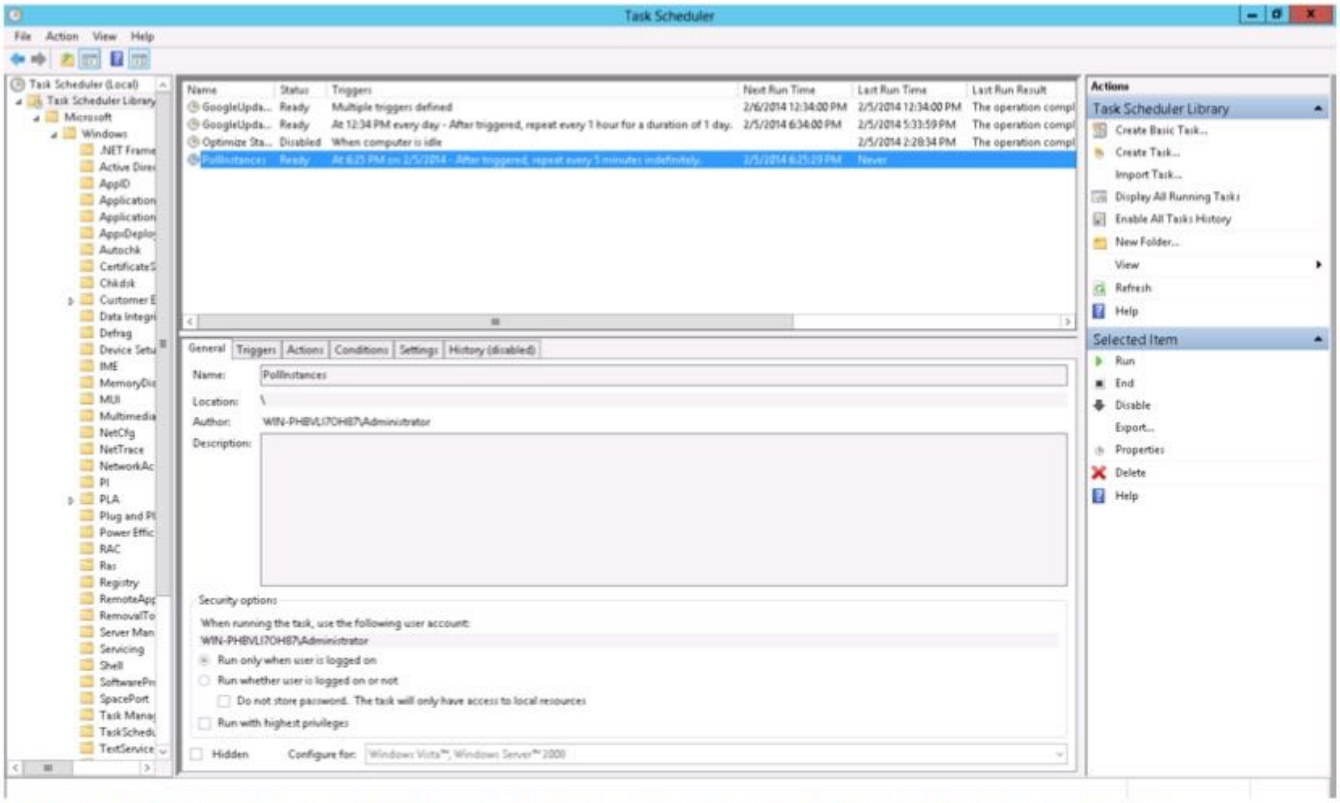


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

85. Add an Action with program set to your `powershell.exe` path and `c:\poll-instances.ps1 <monitoring-server-url>` as argument



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



At the end of this section, you should see your instances updated in the monitoring server every five minutes with nearly the same registration time.

Messaging (SNS and SQS)

In this section you are going to re-generate the monitoring configuration in every auto scaling event that is pushed to an SQS queue.

Create an SQS Queue for monitoring events

86. Navigate to the SQS Management Console
87. Start the **Create New Queue** tool
88. Name and create your queue (e.g. autoScalingEventsQueue) with default settings
89. Open the **Permissions** tab and **Add a Permission**
    Effect: Allow
    Principal: Check "Everybody (*)"
    Actions: Check "All SQS Actions (SQS:*)"
90. From the **Details** tab, copy to a text editor your **Queue ARN** and **URL**

Create an SNS Topic for monitoring events

91. Navigate to the SNS Management Console
92. Start the **Create New Topic** tool and name your topic (e.g. autoScalingEventsTopic)
93. **Create Subscription** for the new topic
    Protocol: Amazon SQS
    Endpoint: [Paste your **Queue ARN**]
94. Navigate to your Auto Scaling Groups
95. Select the auto scaling group created previously (asgMonitoringLab)
96. Open the **Notifications** tab

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

97. **Create Notification**
98. Select the auto scaling notifications topic (e.g. AutoScalingEventsTopic)
99. **Save** your notification

Grant SQS access to your monitoring service

100. Navigate to the IAM Management Console
101. Select **Roles** in the navigation menu
102. Select the previously created role (e.g. MonitoringClient)
103. Open the **Attach Role Policy** tool
104. Select the **Amazon SQS Full Access** policy template

Consume Auto Scaling Event Messages

105. Access your terminal session on the monitoring server instance
106. Download and inspect the script that consumes event messages from SQS:

```
$wc = New-Object System.Net.WebClient

$url = "https://us-east-1-aws-training.s3.amazonaws.com/sys-ops-lab-vpc-
monitoring-windows-v1.1/static/consume-events.ps1"

$wc.DownloadFile($url,"c:\consume-events.ps1")

c:consume-events.ps1 <monitoring-server-url> <queue-url>
```

107. Navigate to your Auto Scaling Groups
108. Select the auto scaling group created previously (asgMonitoringLab)
109. **Edit** your auto scaling group from the **Actions** menu
110. Decrement the **Desired** and **Min** parameter to 1 and **Save**
111. Wait for the event messages to be consumed

At the end of this section you should see event messages in the console and the respective updates in the monitoring service.

Conclusion

Congratulations! You have successfully:

- Written and read custom monitoring statistics from Amazon CloudWatch
- Fired notifications from alarms
- Registered instances dynamically in 3rd party monitoring services

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

End Your Lab

112. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

113. Close any active SSH client sessions or remote desktop sessions.
114. Click the **End Lab** button on the *qwikLAB*™ lab details page.

115. When prompted for confirmation, click **OK**.
116. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS Security - Event Collection Windows Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

Security and network operations, incident management, audit, compliance, and forensics teams rely on logs to understand what is happening and what has happened across their infrastructure. The AWS infrastructure provides logging capabilities, which complement events from operating systems, applications and databases to give additional context when researching potential issues.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with basic Windows server administration and comfortable using the Windows command-line tools.

Topics Covered

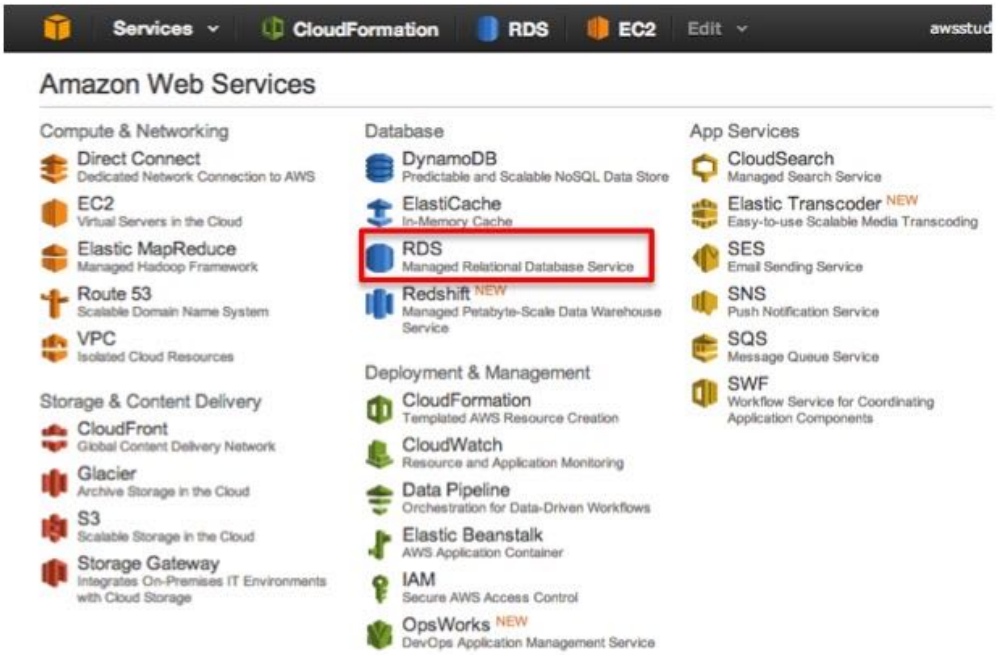
This lab will explore the methods used to obtain the logs from Amazon Relational Database Service (Amazon RDS). The methods used in this lab include:

- Deploying an Amazon RDS instance and Amazon EC2 instance using CloudFormation
- Observing Amazon RDS events in the AWS Console
- Using the command line interface (CLI) to retrieve events from Amazon RDS

Viewing Events

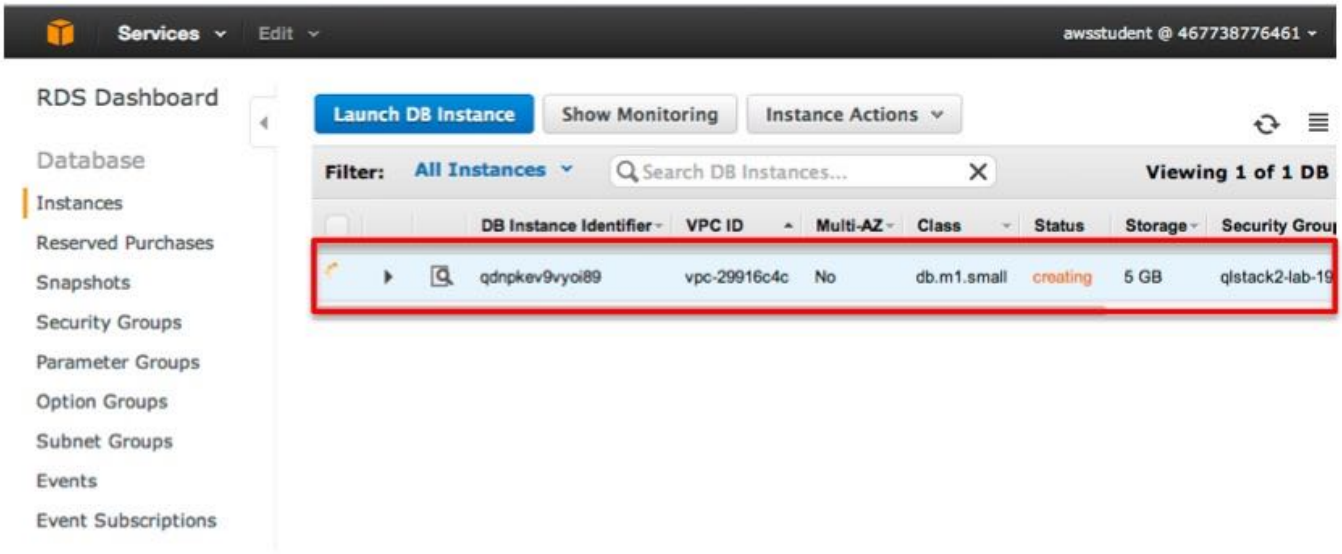
Locating Events with the AWS Console

1. In the AWS Management Console, select "RDS" as shown below.

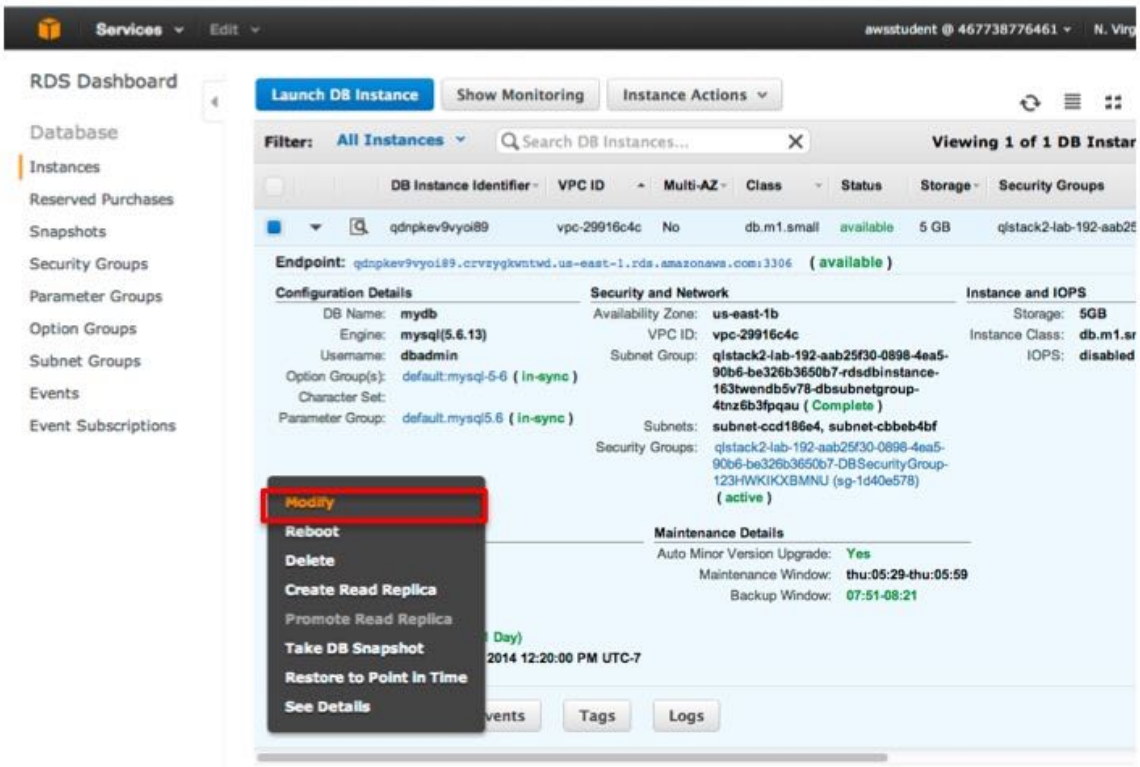


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

2. Click on "Instances" in the left navigation bar. In the RDS Dashboard, you should see a new RDS database instance with a randomized identifier, similar to the screenshot below:

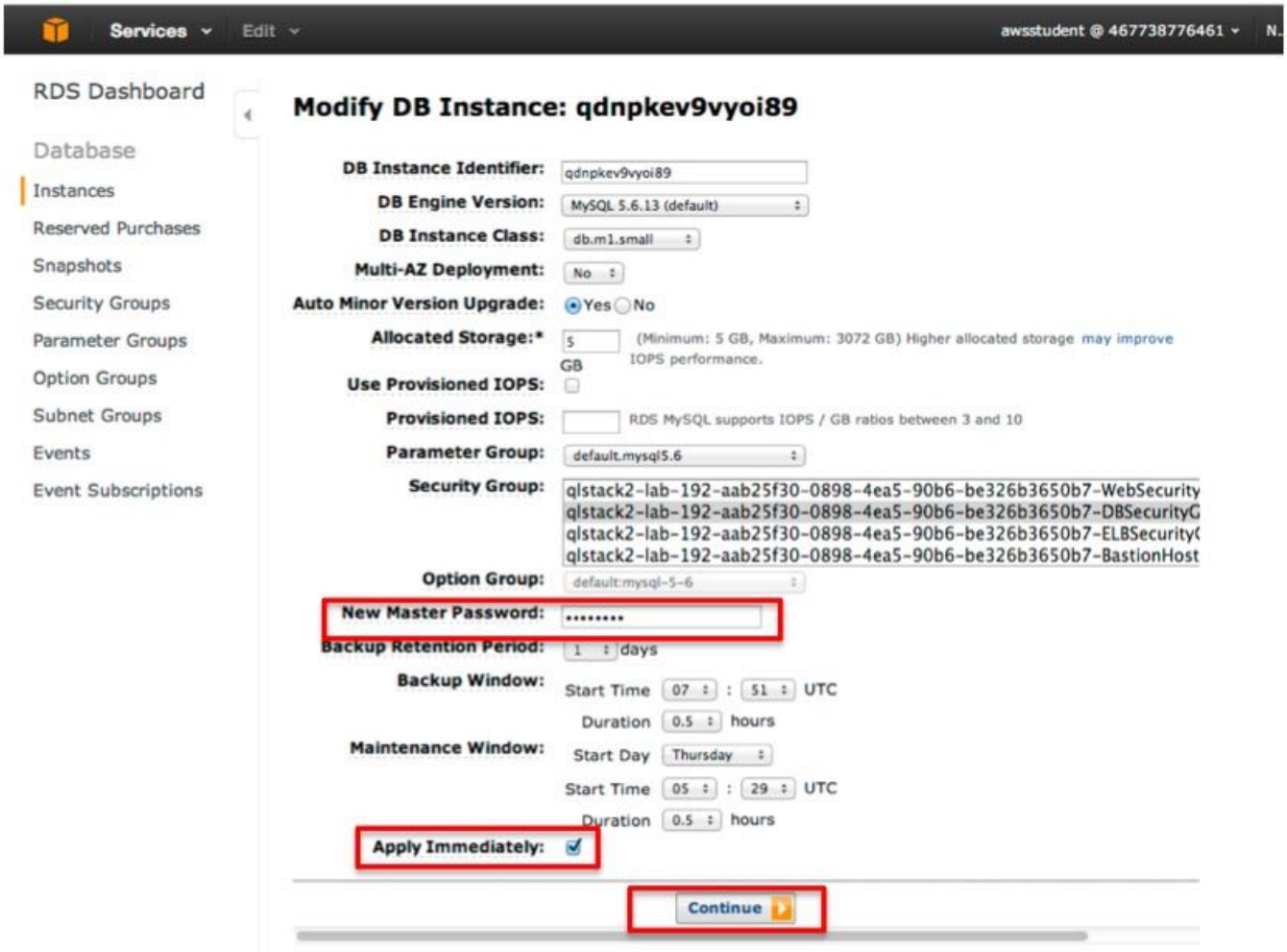


3. If the "Status" reads "Creating" (as shown above), wait until this process completes to proceed. After the database has been created, select the instance, and then choose "Modify" from the "Instance Actions" menu as shown below:



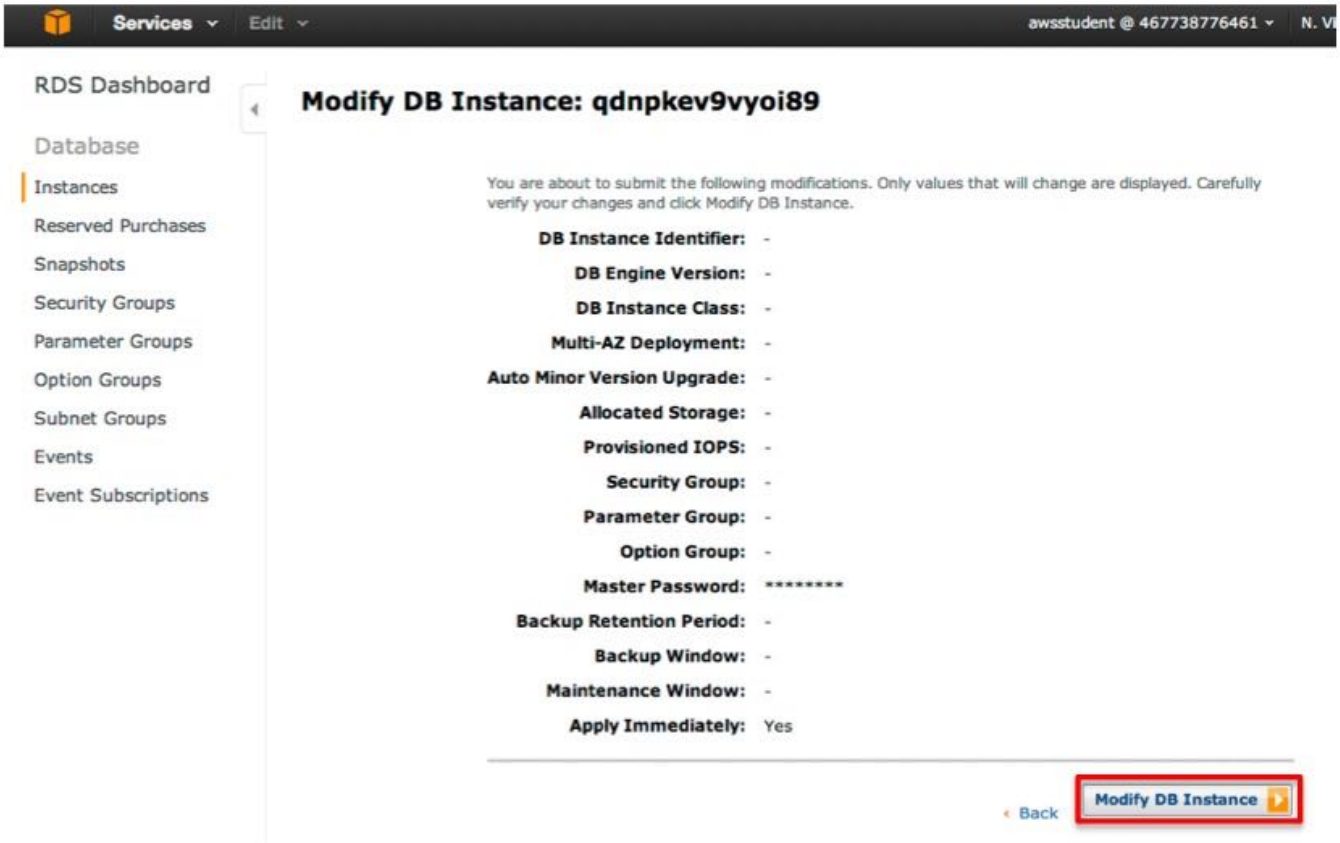
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

4. Now change the password for the database. Provide a new password in the "New Master Password" field, check the "Apply Immediately" box, and click Continue.



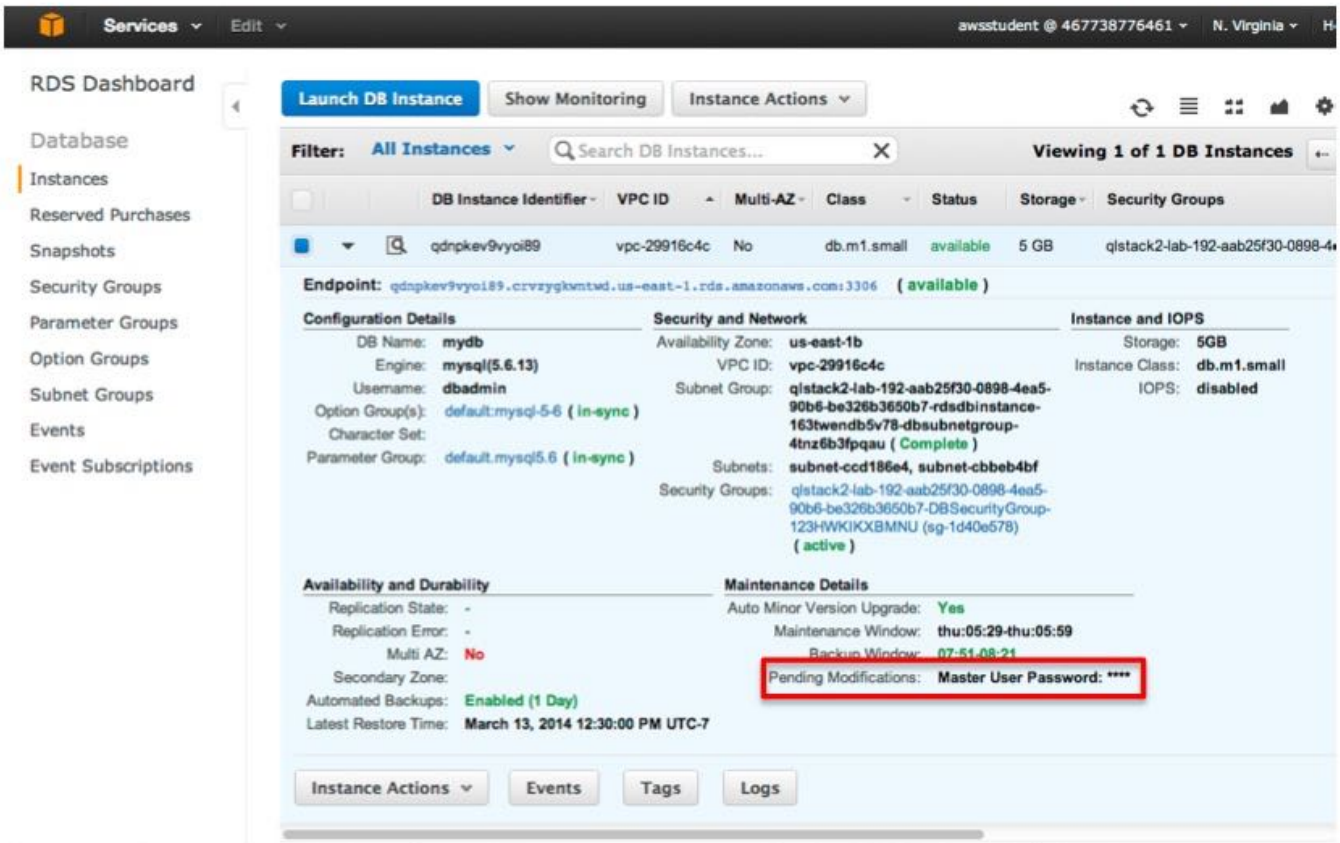
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

5. On the next screen, click "Modify DB Instance" to confirm these changes:



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

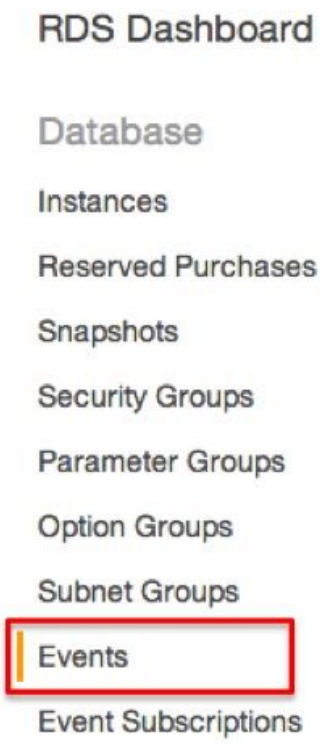
6. When you return to the instance details tab, you should see "Master User Password: ****" in the "Pending Notifications" field as shown below. Wait until this notification clears to proceed.



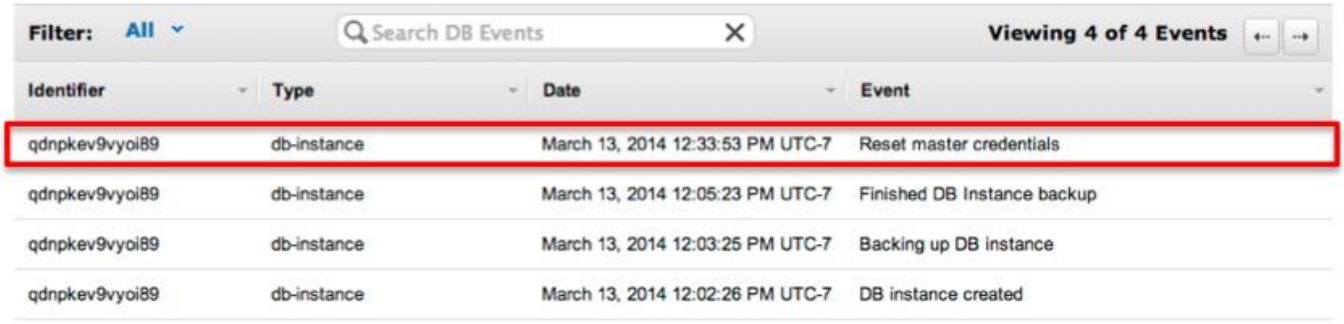
A password change to a database (especially a master password for the RDBMS engine) is typically deemed a significant security event in many organizations. An investigation could be warranted to determine if the change was authorized, and if the change was executed in coordination with change management approval. RDS has an event log, which can be used to provide information to such a query.

7. Click on the "Events" button in the left hand navigation menu of the RDS Dashboard as shown below.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



8. Notice that some events were created when the RDS instance was deployed, including a modification the master credentials.



Identifier	Type	Date	Event
qdnpkev9vyoi89	db-instance	March 13, 2014 12:33:53 PM UTC-7	Reset master credentials
qdnpkev9vyoi89	db-instance	March 13, 2014 12:05:23 PM UTC-7	Finished DB Instance backup
qdnpkev9vyoi89	db-instance	March 13, 2014 12:03:25 PM UTC-7	Backing up DB instance
qdnpkev9vyoi89	db-instance	March 13, 2014 12:02:26 PM UTC-7	DB instance created

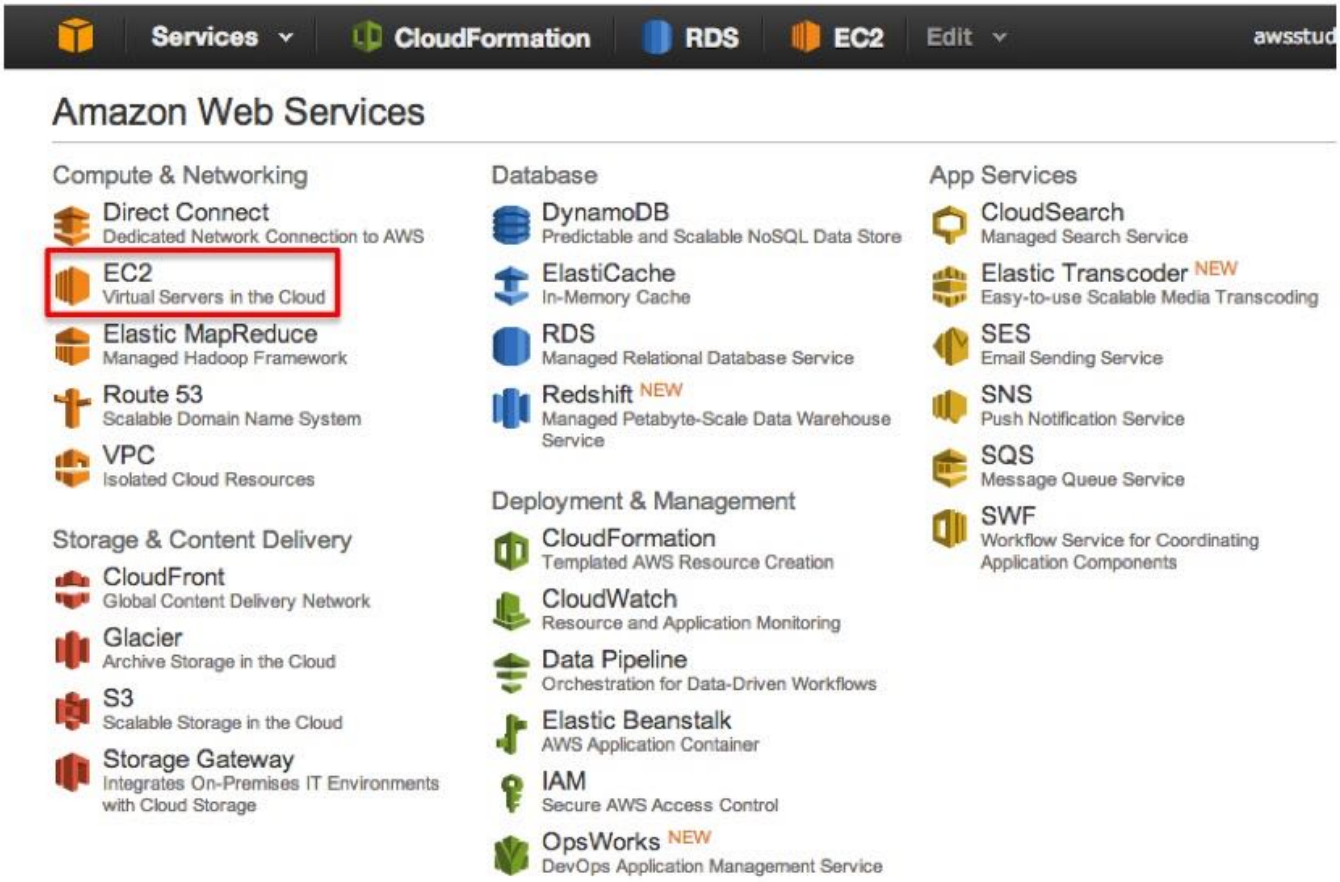
Other services such as Amazon Redshift, CloudFormation, Elastic Beanstalk, and Data Pipeline have similar views of Events in the AWS Console.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Getting Events with the AWS CLI

Events may be also retrieved using the AWS CLI. Programmatic access enables you to periodically collect these logs for analysis. For example, it is useful to send these events to a Security Information and Event Management (SIEM) platform for correlation and analysis by a Security Operations Center (SOC).

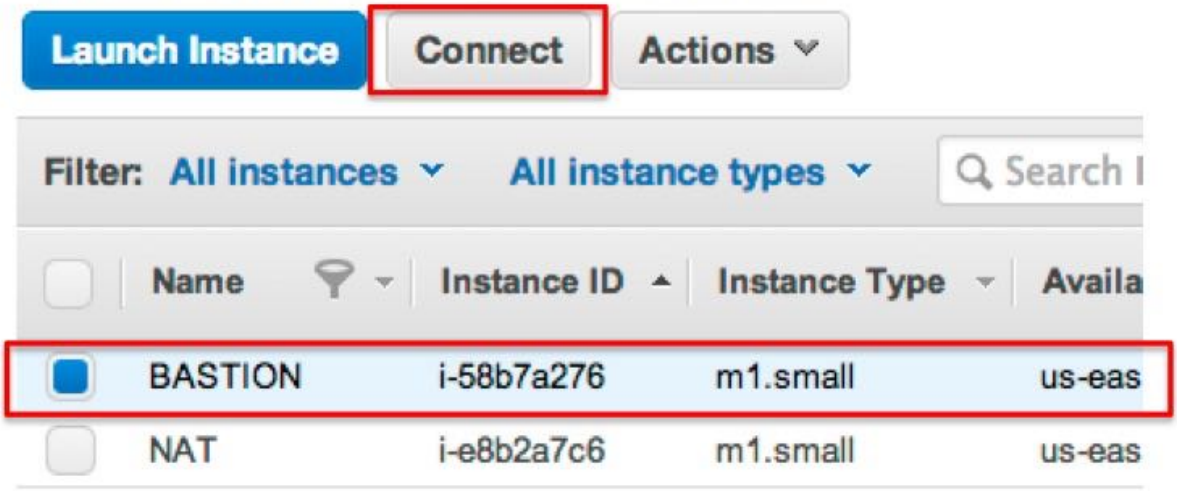
9. An EC2 instance has been launched by this lab. Locate and connect to the EC2 instance as shown below.



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

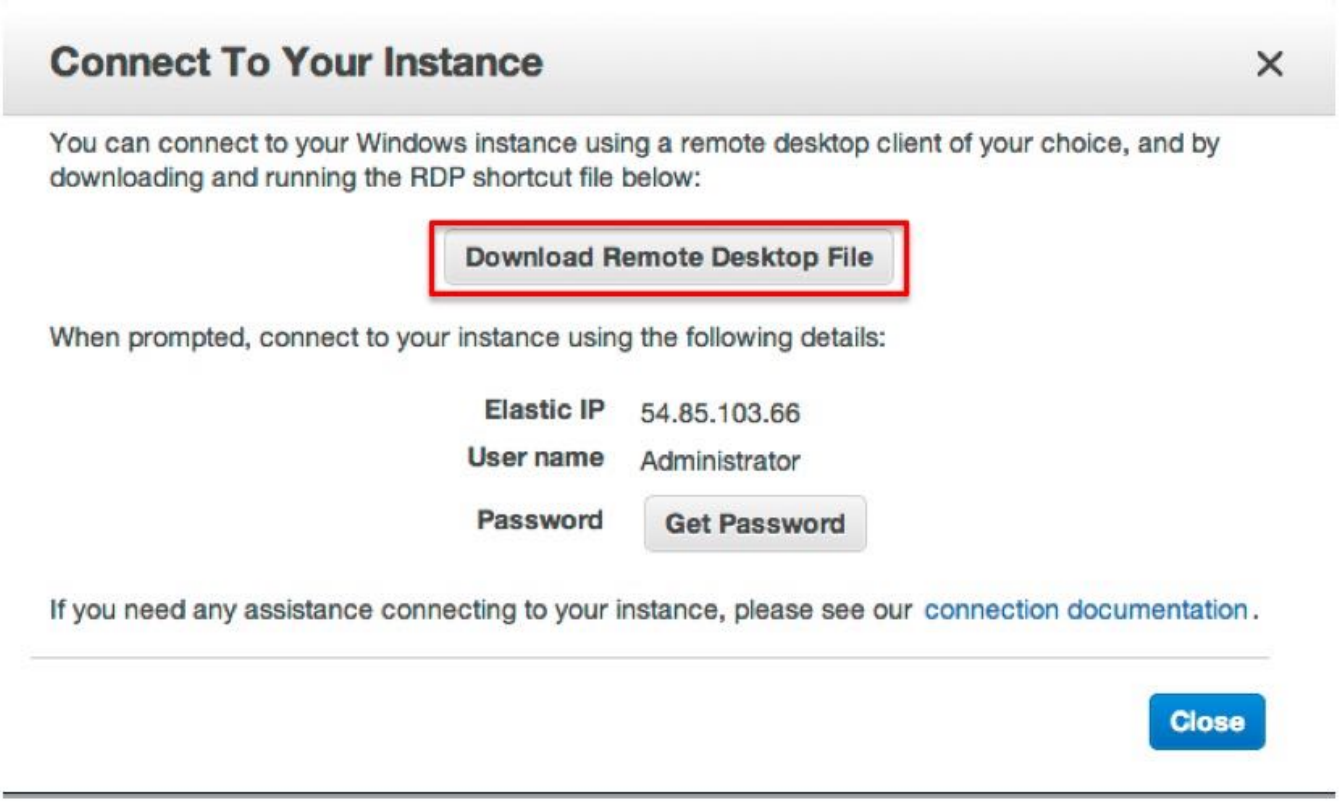
10. Click "Instances" from the left hand navigation menu:

11. Click on the "BASTION" instance, then click the "Connect" button.



12. Click the "Download Remote Desktop File" link to get a preconfigured shortcut for your Remote Desktop Connection. Double-click the downloaded file and connect to the instance. **Note: Your Windows administrator password is the same as your lab user password.**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



13. Open the PowerShell utility on the Windows taskbar[1]. Enter the following command to view RDS events:

```
aws rds describe-events
```

This should produce output similar to as shown below:

[1] Here, we are **using** the PowerShell even though we are not using PowerShell cmdlets. For more information about using AWS Tools for Windows PowerShell, see the appendix below.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
Administrator: Windows PowerShell (4)
PS C:\Users\Administrator> aws rds describe-events
{
    "Events": [
        {
            "Date": "2014-03-13T19:51:22.513Z",
            "Message": "DB instance created",
            "SourceIdentifier": "qd1b6f3mcipuu9q",
            "EventCategories": [
                "creation"
            ],
            "SourceType": "db-instance"
        },
        {
            "Date": "2014-03-13T19:52:52.312Z",
            "Message": "Backing up DB instance",
            "SourceIdentifier": "qd1b6f3mcipuu9q",
            "EventCategories": [
                "backup"
            ],
            "SourceType": "db-instance"
        },
        {
            "Date": "2014-03-13T19:55:39.671Z",
            "Message": "Finished DB Instance backup",
            "SourceIdentifier": "qd1b6f3mcipuu9q",
            "EventCategories": [
                "backup"
            ],
            "SourceType": "db-instance"
        },
        {
            "Date": "2014-03-13T20:00:36.213Z",
            "Message": "Reset master credentials",
            "SourceIdentifier": "qd1b6f3mcipuu9q",
            "EventCategories": [
                "configuration change"
            ],
            "SourceType": "db-instance"
        }
    ]
}
```

Events can also be collected through the AWS API. For more information, visit http://aws.amazon.com/documentation/

Conclusion

Congratulations! You now have successfully:

- Viewed events for Amazon RDS in the AWS Console
- Viewed events for Amazon RDS using the AWS CLI

For feedback, suggestions and corrections to this lab, please email aws-course-feedback@amazon.com.

Appendix: Using PowerShell cmdlets

Optionally, you can poll the AWS API for events using the Windows Commandlets. The AWS Tools for Windows PowerShell are a set of PowerShell cmdlets that are built on top of the functionality exposed by the AWS SDK for .NET. The AWS Tools for Windows PowerShell enable you to script operations on your AWS resources from the PowerShell command line. Although the cmdlets are implemented using the service clients and methods from the SDK, the cmdlets provide an idiomatic PowerShell experience for specifying parameters and handling results. For example, the cmdlets for the PowerShell Tools support PowerShell pipelining—that is, you can pipeline PowerShell objects both into and out of the cmdlets.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

The AWS Tools for Windows PowerShell are flexible in how they enable you to handle credentials including support for the AWS Identity and Access Management (IAM) infrastructure; you can use the tools with IAM user credentials, temporary security tokens, and IAM roles.
The AWS Tools for Windows PowerShell support the same set of services and regions as supported by the SDK.
For this lab, open the PowerShell utility on the Windows taskbar. Enter the following command, using the values you captured earlier for AccessKey and SecretKey.
Tip: Use a text editor to build your command then paste it into the PowerShell window.

```
Get-RDSEvent -Region [yourregion] -AccessKey
AKIAIOSFODNN7EXAMPLE -SecretKey
wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

This should produce output similar to as shown below:

```
SourceIdentifier : rds-win-rds-1p6ypwr872u7d-dbsecuritygroup-16rfc3n5tabif
SourceType       : db-security-group
Message          : Applied change to security group
EventCategories  : {configuration change}
Date             : 7/3/2013 4:57:28 PM

SourceIdentifier : rd17q4faclj8hqw
SourceType       : db-instance
Message          : DB instance created
EventCategories  : {creation}
Date             : 7/3/2013 5:00:42 PM

SourceIdentifier : rd17q4faclj8hqw
SourceType       : db-instance
Message          : Backing up DB instance
EventCategories  : {backup}
Date             : 7/3/2013 5:01:30 PM

SourceIdentifier : rd17q4faclj8hqw
SourceType       : db-instance
Message          : Finished DB Instance backup
EventCategories  : {backup}
Date             : 7/3/2013 5:03:46 PM
```

You can use the PowerShell `Get-Command` cmdlet to obtain a list of all of the cmdlets that are provided by tools.

End Your Lab

14. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

15. Close any active SSH client sessions or remote desktop sessions.
16. Click the **End Lab** button on the *qwikLAB*™ lab details page.

17. When prompted for confirmation, click **OK**.
18. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS Logging Windows Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

Although AWS EC2 instances are meant to be dynamic by design, the application or system logs on those resources may contain valuable information that should be preserved for potential later analysis. In this lab, we will walk through the process of creating and implementing an S3 bucket for aggregated log content.

In this lab you will:

- Create an S3 destination bucket for system logs
- Modify an IAM role to allow access to the S3 log destination
- Create a key schema for log prefixes
- Write and test a log rotation script to send logs to S3 from your centralized logging server

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with EC2 instances and using RDP to access EC2 instances.

Topics Covered

This lab will focus on the creation of a script utilizing the AWS CLI. It will briefly touch on S3 bucket creation, S3 bucket lifecycle policies, and IAM Roles.

Connect to the Bastion Host

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair

This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this guide called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group

It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

1. Retrieve your current public IP address by navigating your browser to http://ip.appspot.com
2. Open the **AWS Management Console**.
3. Select the **VPC** service.
4. From the menu on the left side, click **Security Groups**.
5. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion Host"**.
6. On the bottom pane, click the **Inbound** tab.
7. From the **Create a new rule dropdown**, select RDP.
8. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
9. Click the **Add rule** button.
10. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host

11. Open the **AWS Management console**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

12. Select the **EC2** service.
13. From the menu on the left side, click **Instances**.
14. **Select the BastionHost instance** by clicking on the box next to its name.
15. The Public IP is displayed on the bottom pane. Please write it down.

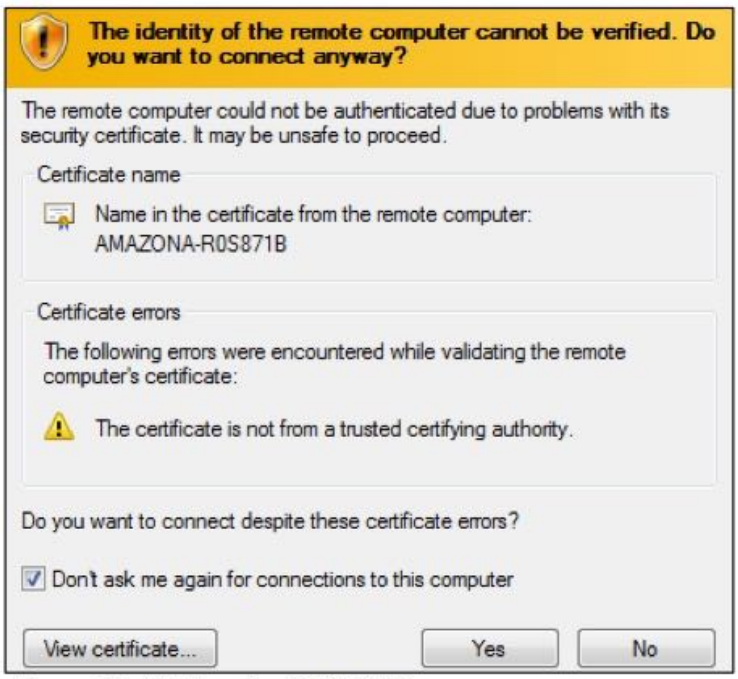
Retrieve the bastion instance password
Before connecting to the bastion using RDP, you must retrieve the instance's password for the Windows Administrator account.
16. In your browser, switch to the **qwikLAB** tab.
17. In the **Lab Connection** portion of the page, copy the **Password** in the **RDP** section – you may also paste it into a text file for later retrieval.

Note: This is the password you use to log into the bastion instance using the remote desktop client.
18. If the computer you are using to complete this lab is Windows-based, proceed to the next section. If you are using an OS X or Linux-based machine, proceed to the section appropriate to your operating system.

Connecting to the bastion instance (Windows)
Important: Proceed to one of the later sections entitled "Connect to the bastion instance (OS X)" or "Connect to the bastion instance (Linux)" if you are using a Linux-based laptop to perform this exercise. This section is *only* for Microsoft Windows users. There is no need for you to perform the steps in all sections. Only complete the section that matches your operating system.
19. On your local computer click **Start > Run**, type **MSTSC** and click **OK** to start the local RDP client.
20. Type or paste the Elastic IP address of the bastion instance in the **Computer** field.
21. Click **Show Options**, and type **\Administrator** in the **User name** field.

Note: Specifying the user name as "\Administrator" (with a leading backslash) differentiates the bastion Administrator account from the Administrator account on your local computer.
22. Click **Connect**.
23. When prompted, type the RDP **Password** you noted then click **OK**.
24. Click **Yes** if you see a certificate verification message similar to "the identity of the remote computer cannot be verified:"

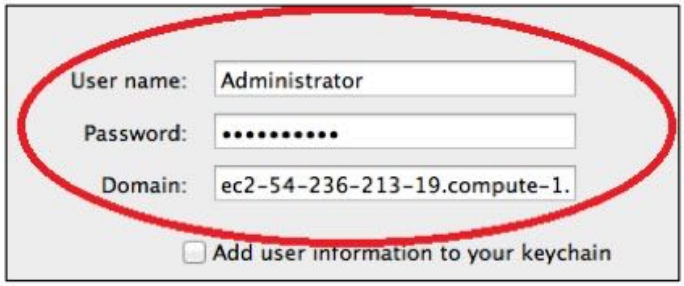
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



25. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (OS X)
Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." This section is for OS X users *only*. If you are using a different Linux-based operating system, please proceed to the section entitled "Connect to the bastion instance (Linux)." There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

26. Open the Remote Desktop Connection for Mac application.
27. Type the bastion instance's IP address you noted in the **Computer** field and click **Connect**.
28. When prompted, in the **User name** field, type Administrator and then type the **Password** you noted from the qwikLAB page. The Domain should be automatically set to the EC2 Instance DNS name, and you can ignore it.



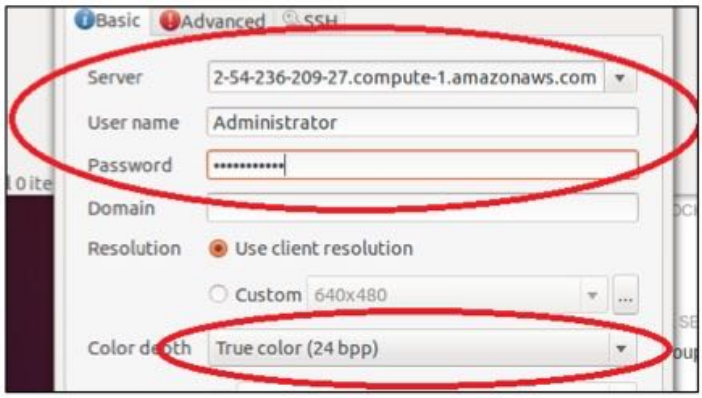
29. Click **OK**.
30. Click **Connect** if you see a verification message similar to "the server name is incorrect."
31. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (Linux)
Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." Only complete this section if you are using a non-

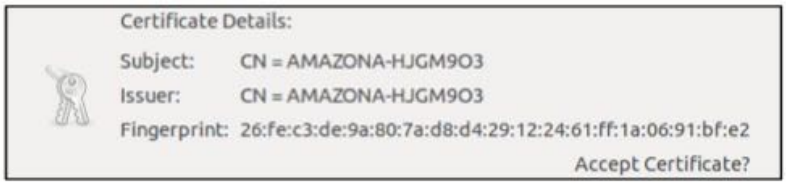
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Macintosh, Linux-based laptop to perform this exercise. This section is *only* for Linux users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

32. Open the Remmina Remote Desktop Client.
33. Type the bastion instance's IP address you noted in the **Server** field.
34. In the **User name** field type **Administrator**, and in the **Password** field, type the password from the qwikLAB page.
35. Optionally, choose a **Color depth** that your bandwidth supports (in this example 'True color (24 bpp)') for a nicer remote desktop environment.



36. Click **Connect**.
37. Click **OK** when prompted to accept the remote certificate.



Create S3 log destination and allow access from the log server

Create S3 log destination bucket

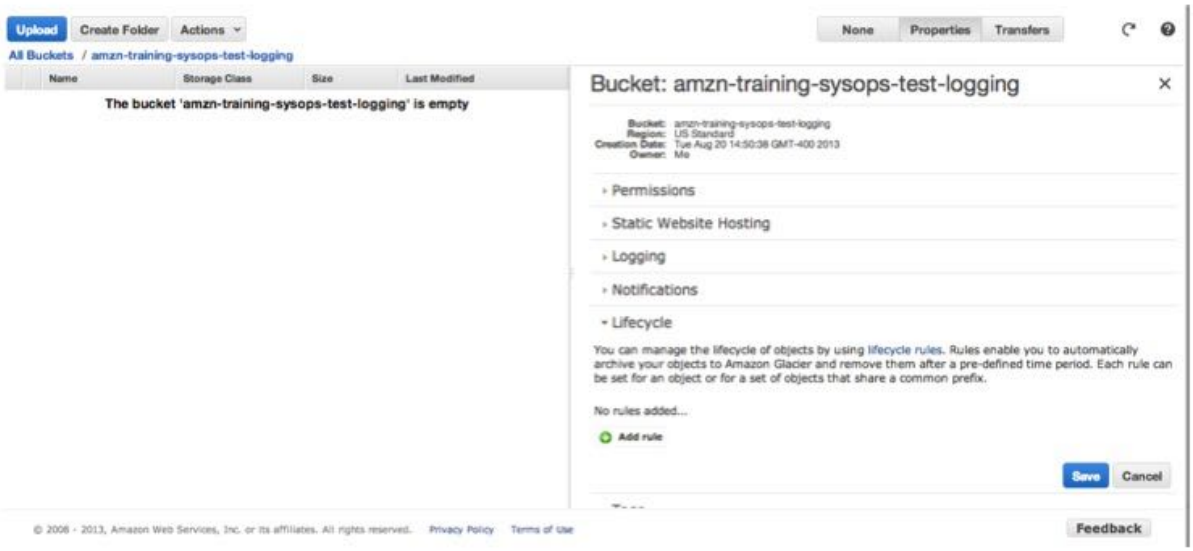
Log into the AWS Management Console and navigate to **S3.** Click on the "Create Bucket" button. Give your new bucket a meaningful name, something like "[yourcompanyname]-test-logging" - keep in mind that bucket names are globally unique, so you won't be able to create a bucket called "logging". Select the region that is closest to your primary EC2 infrastructure. If you plan to implement this in multiple regions, you might consider creating one bucket in each region.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Set bucket lifecycle policy

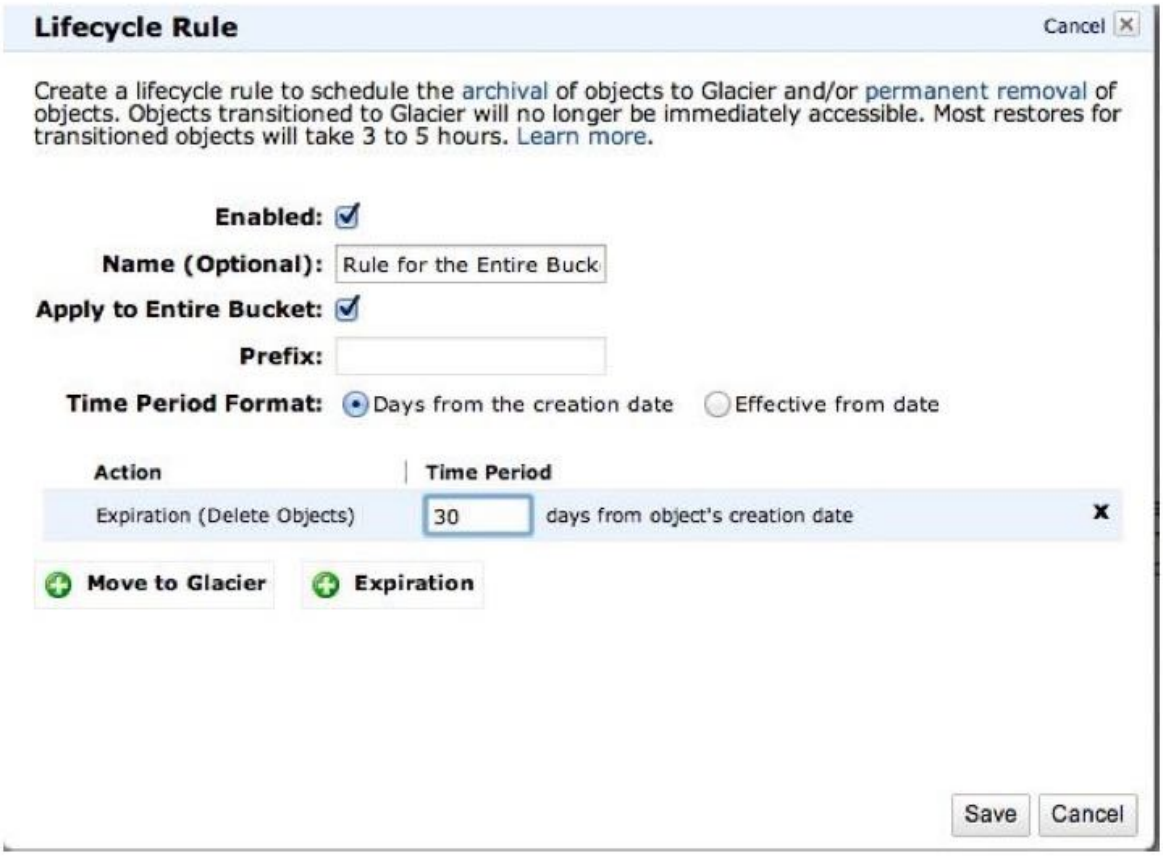
Since logs are generally only useful for a limited time, this bucket should have a lifecycle policy to periodically delete older objects. In this lab, we will set the lifecycle of all log files at 30 days.

1. Select the new bucket in the S3 Console and click the "Properties" button at the top-right of the console. Click on "Lifecycle" on the right and then click on the "Add rule" button.



2. In the resulting screen, check the box marked "Apply to Entire Bucket", then click "Expiration". Enter "30" in the text box for "Time Period" and click the "Save" button. Click "OK" on the notification.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

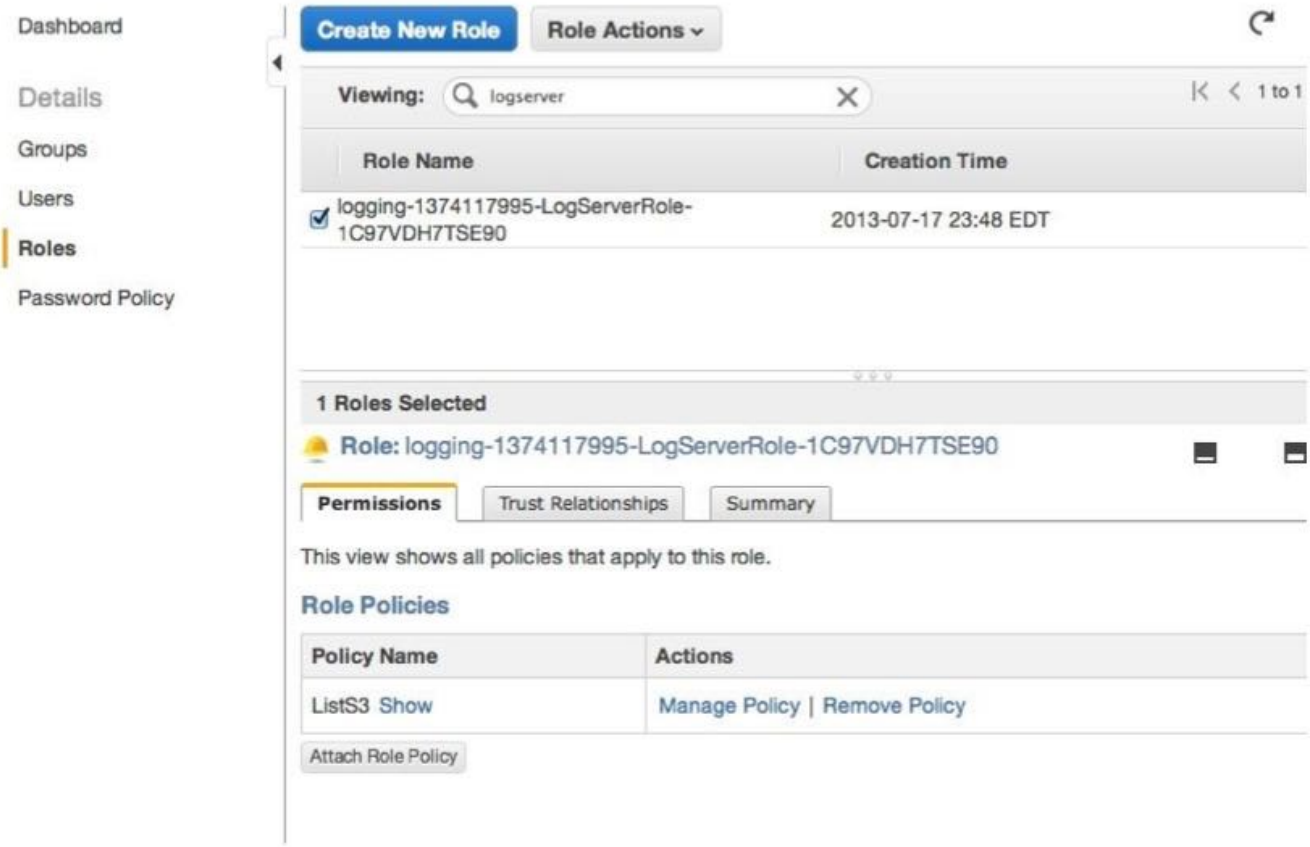


Create IAM Role Policy

The bucket is currently only accessible when accessing the S3 API or AWS Management Console with credentials. Since it needs to be accessible by the log server without credentials, we need to enable the log server's IAM Role to access the S3 bucket. This can be done in one of 2 ways: modify the bucket policy to allow access from the IAM Role, or modify the IAM Role's policy to be able to access the bucket. In this lab, we are going to modify the IAM Role's policy to be able to access the bucket.

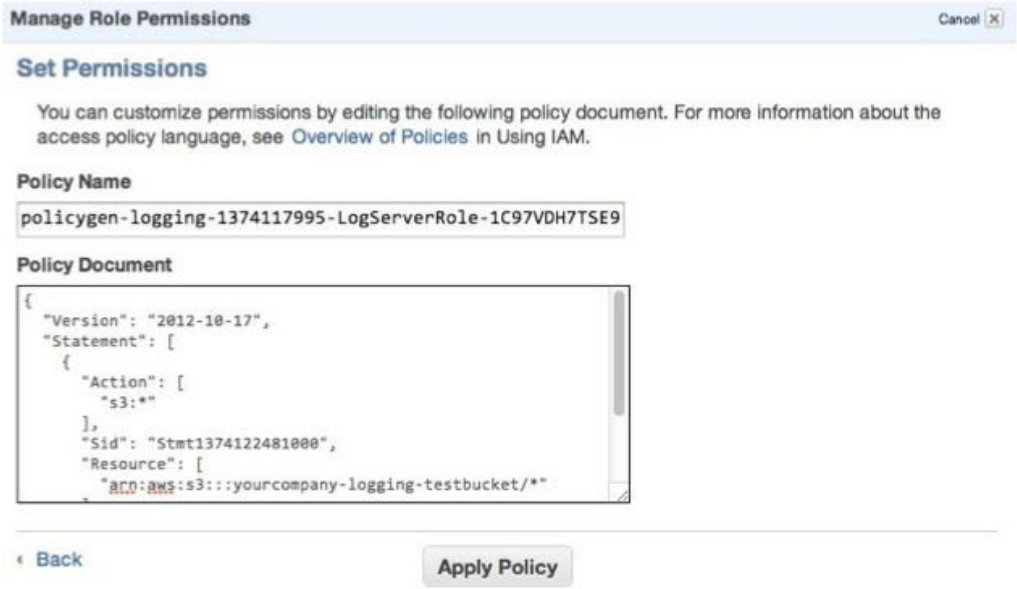
3. Open a new AWS Management Console window and navigate to **IAM.** Select "Roles" from the menu on the left.
4. Search for the string "logserver" – one role should match. Select that role and open the tab "Permissions" in the bottom half of the console.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



5. Under Role Policies, there should only be one policy, called "root". Click on the button at the bottom labeled "Attach Role Policy".
6. Click the second radio button labeled ""Policy Generator", and click "Select".
7. Leave "Effect" as "Allow"
8. Select "Amazon S3" from the "AWS Service" dropdown
9. In the "Actions" dropdown, check the box for "All Actions"
10. In the "Amazon Resource Name" text box, enter "`arn:aws:s3:::<logging-bucket-name>*`", replacing **`<logging-bucket-name>`** with the name of your S3 bucket.
11. Click "Add Statement" and click "Continue"
12. The resulting JSON policy will be displayed, along with a suggested name. The policy name can be changed without impacting any functionality.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



13. Click "Apply Policy"
14. After modifying this policy, the log server should now have access to the S3 bucket.

Log into logging server and test S3 access

In order to log into the logging server, you will first need to log into the bastion host and subsequently the logging host (see Appendix A and the appropriate "Connect to the Windows Bastion Server" and "Connect to a Windows Instance…" segments).

15. Right click the PowerShell icon on the taskbar and run it as administrator.
16. Get a list of s3 buckets by entering the following command:

```
aws s3 ls
```

17. You should see your logging bucket in the list.
18. To confirm that you have write access, create a small file in a temporary directory and try uploading it to the bucket:
19. Replace **<your-bucket-name>** with the name of your bucket.

```
echo "hello" > C:\aws\bucketTest.txt
aws s3 cp C:\aws\bucketTest.txt s3:// <your-bucket-name>/
```

20. Refresh your S3 console to confirm that the object was uploaded.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Enable script execution on the system

21. We will automate the archiving process with a PowerShell script. As you're probably aware, the simplest approach is to enable script execution on the system. Enter the following in the PowerShell window and confirm the change when prompted:

```
Set-ExecutionPolicy RemoteSigned
```

Check the staging folder for logs

22. Confirm that the folder Z:\application_logs exists and has some log files in it.

Create log archive script

23. Using the same PowerShell shortcut on the taskbar, run ISE as administrator and save the following script to "C:\aws\archive-syslogs.ps1". Replace **<your-bucket-name>** with the name of your bucket.

```
<#
.SYNOPSIS
    Archives files to S3 optionally using a regex file pattern
.DESCRIPTION
    If a pattern with is provided, only matching files are archived,
otherwise
    everything in a folder is uploaded
.NOTES
    Author : awsstudent
#>

[CmdletBinding()]
Param(
    [Parameter(Mandatory=$false)][ValidateScript({Test-Path $_ -
PathType 'Container'})][string]
    $Path = "Z:\application_logs",
    [Parameter(Mandatory=$false)][string]
    $Bucket = "<your-bucket-name>",
    [Parameter(Mandatory=$false)][string]
    $Pattern = ".*"
    )

Get-ChildItem $Path |
    Where-Object { $_.Name -match $Pattern -and ! $_.PSIsContainer -
and $_.Length -gt 0 } |
    Select-Object Name, FullName | ForEach-Object {
        aws s3 cp $_.FullName "s3://$Bucket/$($_.Name)"
        Remove-Item -Path $_.FullName
    }
```

24. There should already be plenty of log data to test with, so try running the script from the PowerShell window:

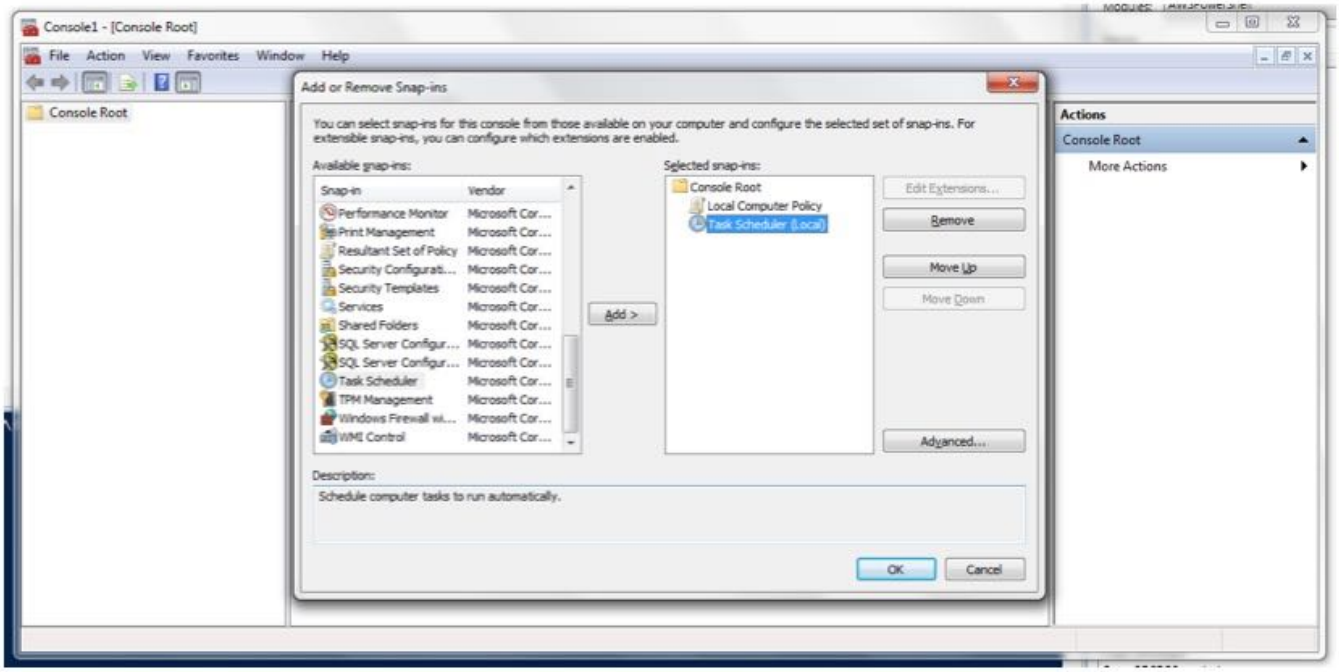
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
C:\aws\archive-syslogs.ps1
```

25. Refresh the S3 console to confirm that the log files were uploaded, and check the directory to confirm that they have been deleted.

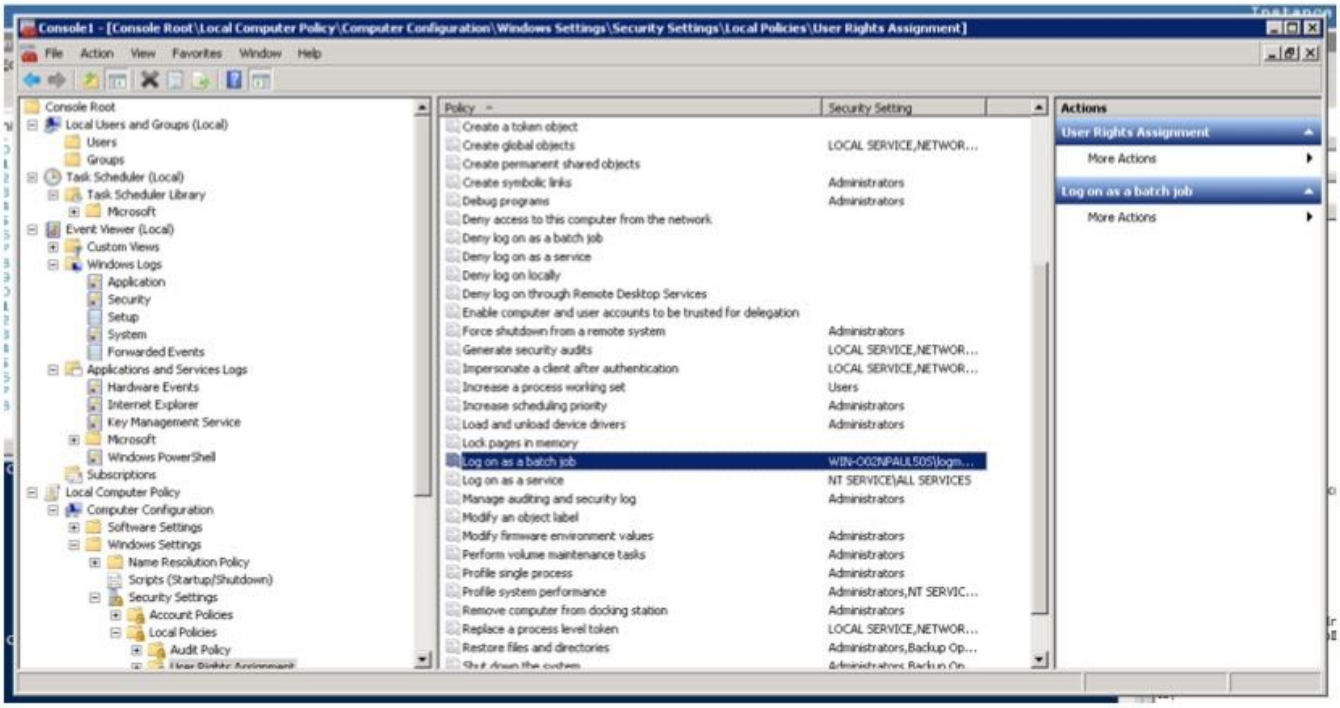
Create a scheduled task to periodically archive log files

26. Let's open the management console and add the two snap-ins we'll need next. To do this, use **Run > "mmc"** and go to **File > Add / Remove Snap-ins**. Select *Group Policy Object Editor* and *Task Scheduler (Local)* from available snap-ins. When prompted, choose to add the respective snap-in for the local machine.

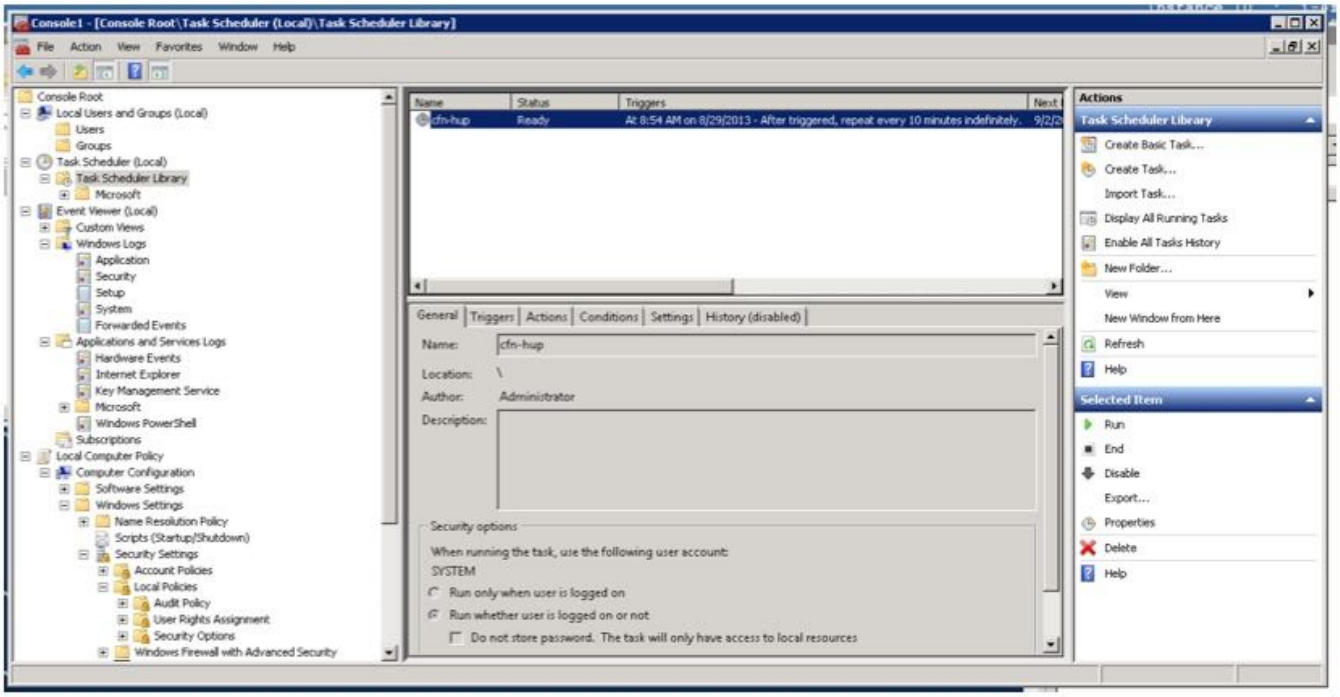


27. We already have an account dedicated to logging, so we'll leverage it for the scheduled archive task. First, we'll need to apply the "`Log On As Batch Job`" permission. You can do this in **MMC > Local Computer Policy > Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



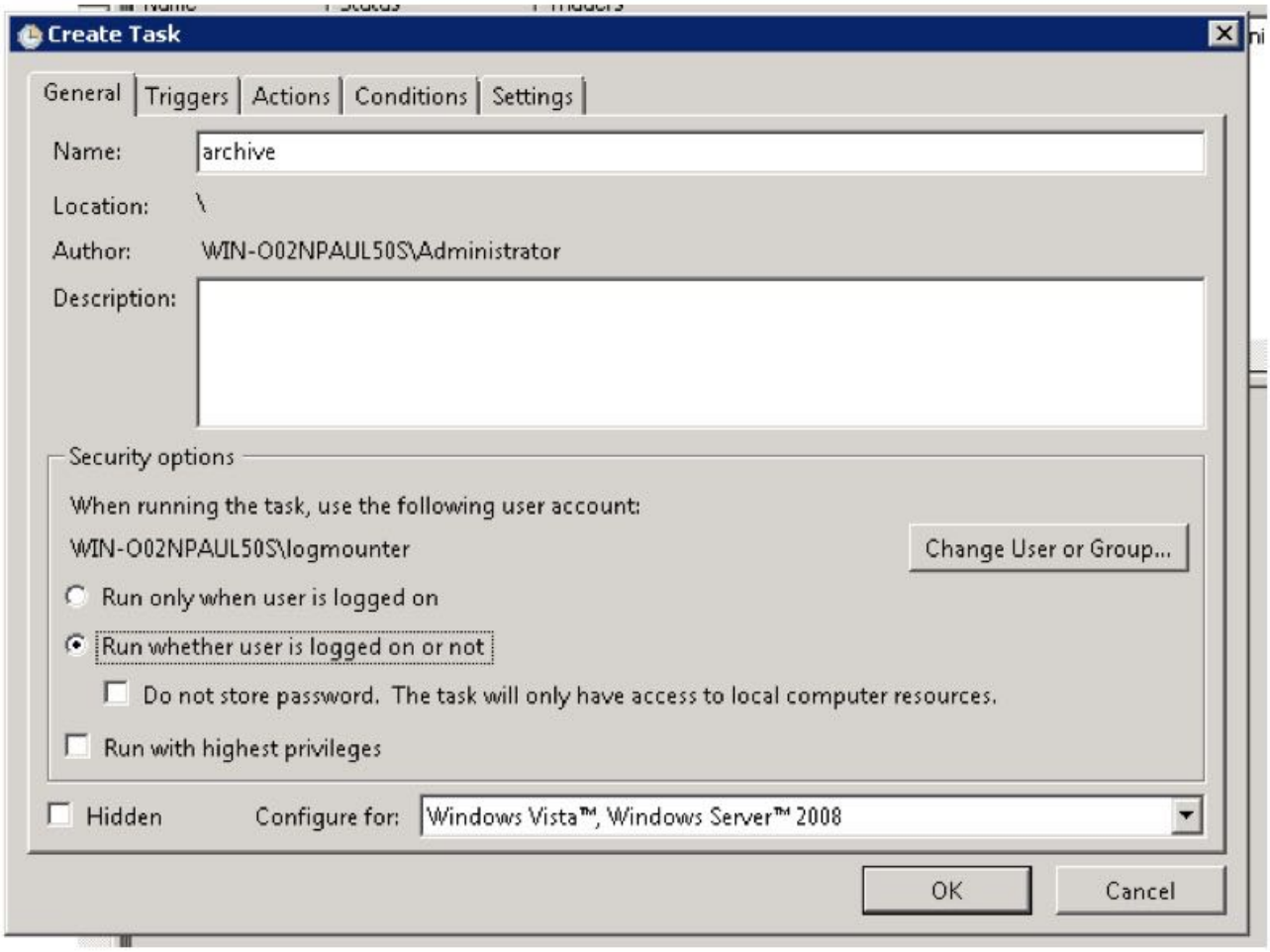
28. Next, navigate to **MMC > Task Scheduler (Local) > Task Scheduler Library**.



29. Create a new scheduled task from the action pane on the right or by right-clicking in the task list table area. Name it "`archive`" and have the user "`logmounter`" run it by using the "`Change`

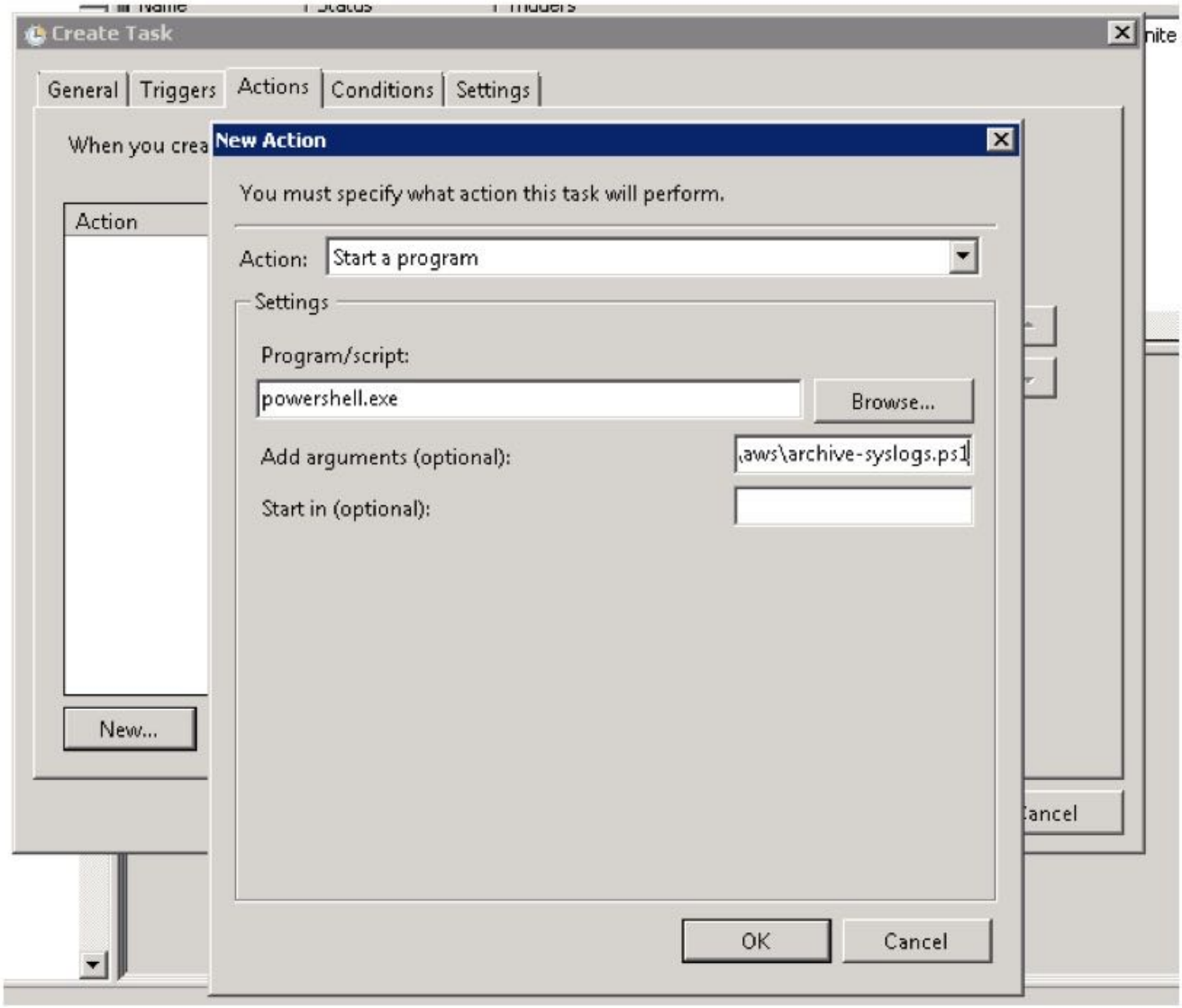
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

User or Group…" button on the general tab. Be sure to check "Run whether user is logged in or not…".



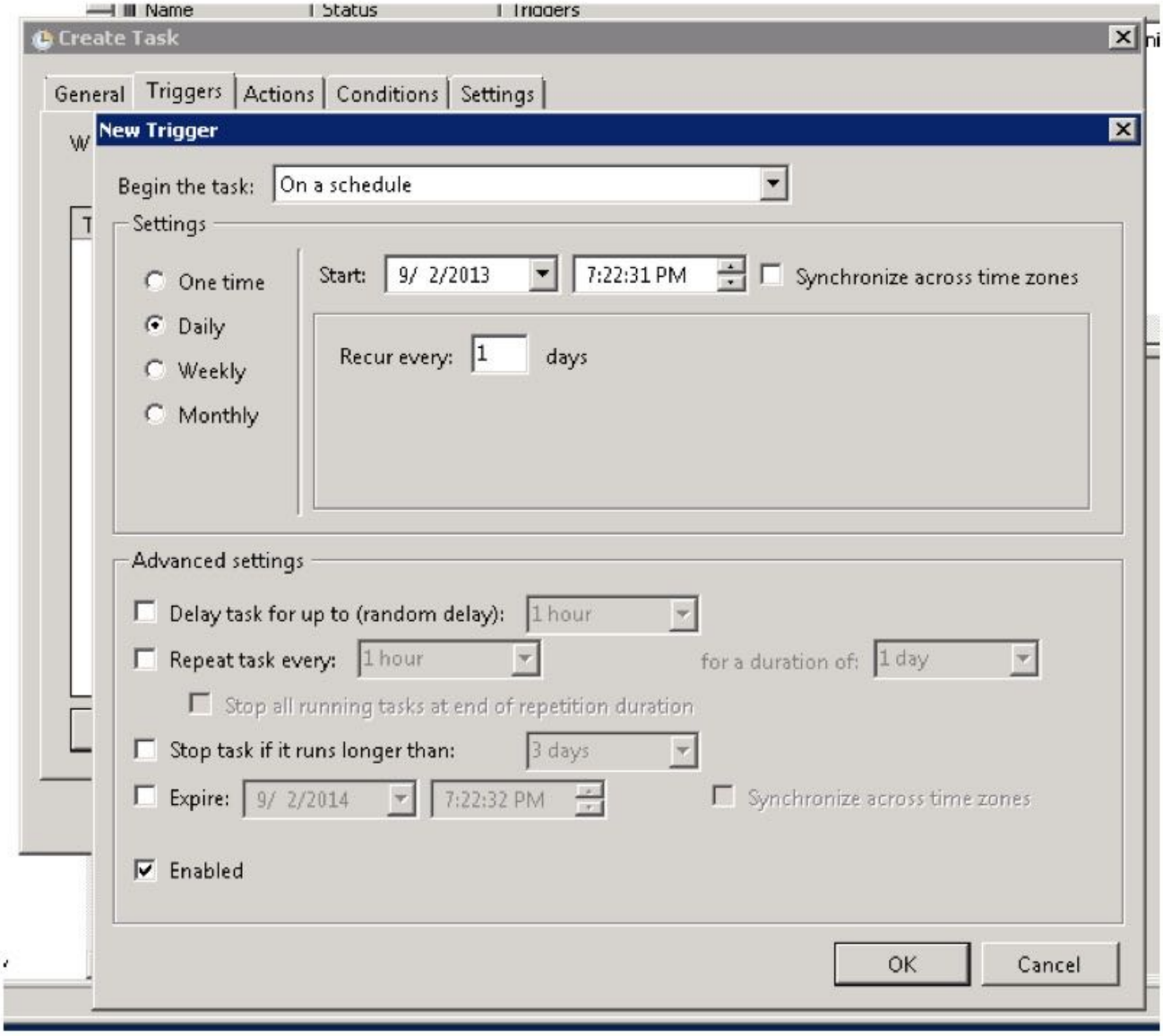
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

30. On the **Actions** tab, add an action to run the script. The "`Program`" is "`powershell.exe`" and the "`Arguments`" are "`-file C:\aws\archive-syslogs.ps1`".



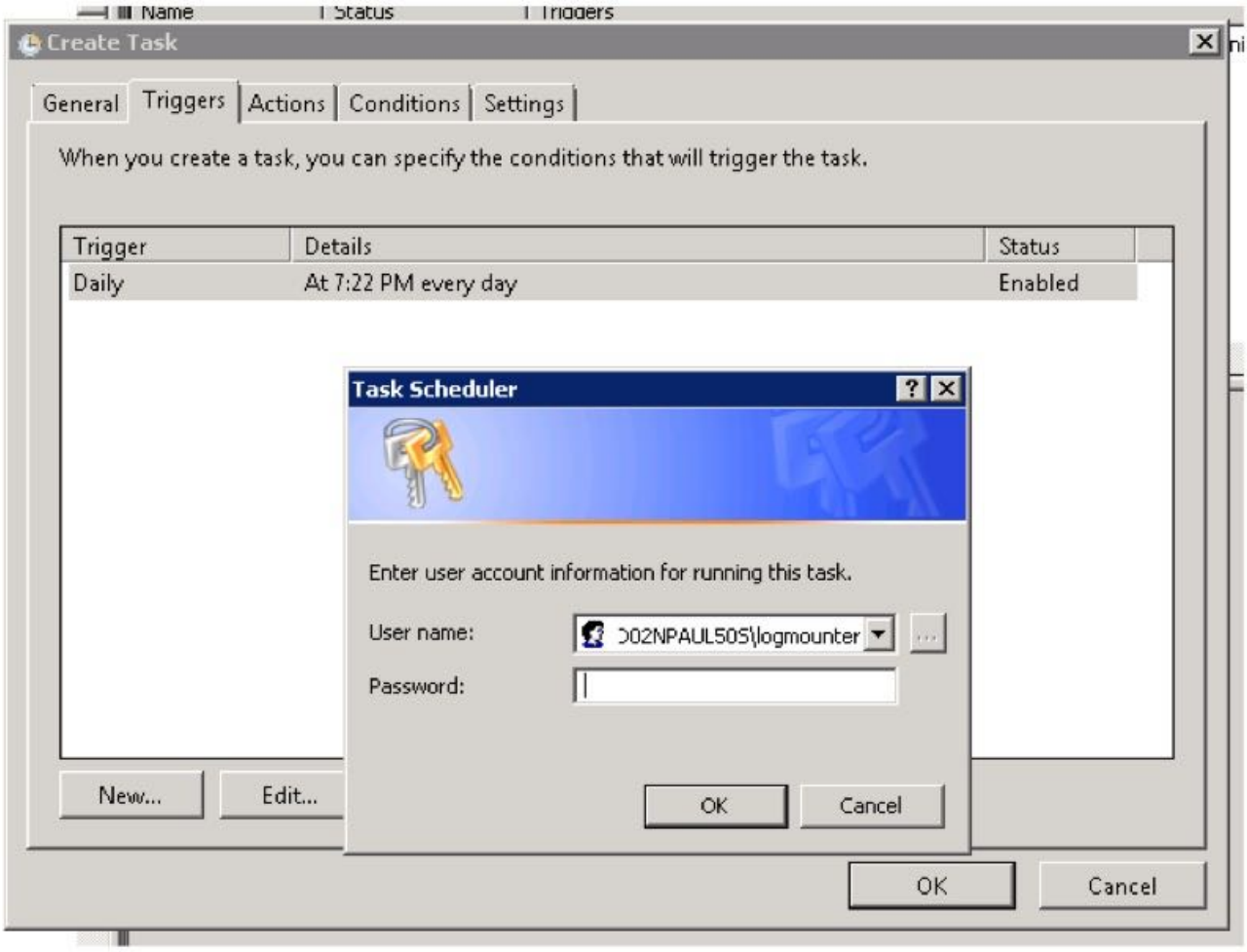
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

31. On the **Triggers** tab, set up the task to run daily (default time settings will serve).



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

32. When you hit ok to save the task, you'll be prompted for the user password, "`iM0untlaWGS`".



Manually run the archive scheduled task

33. Test that the scheduled task will perform its job by right-clicking it in the table and choosing "`Run`". After it completes, the system logs should have been moved to your S3 bucket.

Improve to support compression and key distribution

At this point, everything is functional – there are two improvements we can (and should) make.

34. Compress logfiles before uploading them to S3. We can use a faster compression than default for efficiency if desired.
35. Create a key prefix with a date stamp.

Both of these changes can be implemented in the `C:\aws\archive-syslogs.ps1` script as follows:

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
<#
.SYNOPSIS
    Archives files to S3 in a zip file optionally using a regex file
pattern
.DESCRIPTION
    If a pattern with is provided, only matching files are archived,
otherwise
    everything in a folder is zipped and uploaded
.NOTES
    Author : awsstudent
#>

[CmdletBinding()]
Param(
    [Parameter(Mandatory=$false)][ValidateScript({Test-Path $_ -
PathType 'Container'})][string]
    $Path = "Z:\application_logs",
    [Parameter(Mandatory=$false)][string]
    $Bucket = "<your-bucket-name>",
    [Parameter(Mandatory=$false)][string]
    $Pattern = ".*"
    )

$logs = Get-ChildItem $Path | Where-Object { $_.Name -match $Pattern -
and ! $_.PSIsContainer -and $_.Length -gt 0 } |
    Select-Object Name, FullName

if (!($logs | Measure-Object).Count) { Write-Warning "no files at
$Path" -WarningAction Stop}

$archive = (Join-Path (Resolve-Path $Path) "logs.zip")
Set-Content $archive ("PK" + [char]5 + [char]6 + ("$([char]0)" * 18))
$ns = (New-Object -com shell.application).NameSpace($archive);

$logs | ForEach-Object { $ns.MoveHere($_.FullName); sleep 1 }

aws s3 cp $archive "s3://$Bucket/$(Get-Date -UFormat
"%Y/%m/%d/%H/%M")/logs.zip"
Remove-Item $archive
```

36. Trigger the task again, and check your S3 bucket to confirm that the scripts are working as expected.

Epilogue and bonus

Congratulations, you have set up log rotation to S3!
Bonus activity (answer on next page): determine how you could ensure that all logs go to S3 if an instance is removed from an Auto Scaling group (hint: Auto Scaling will issue a shutdown command on the instance when it is removed from the Auto Scaling group)

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Bonus activity: The trick here is to force a log rotation before shutting down. The best way to do this is to create another scheduled task to run at shutdown. Here is one way to do that:

1) In your MMC windows, navigate to **MMC > Local Computer Policy > Computer Configuration > Windows Settings > Scripts (Startup/Shutdown) > Shutdown**
2) Ensure that you are on the tab labeled "Scripts"
3) Click "Add"
    a. Under "Script Name" enter "powershell.exe"
    b. Under "Script parameters" enter "-file C:\aws\archive-syslogs.ps1"
    c. Click "OK"
    d. Click "OK"

Test it out by removing all logs from the S3 bucket and rebooting the instance.

End Your Lab

37. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

38. Close any active SSH client sessions or remote desktop sessions.
39. Click the **End Lab** button on the *qwikLAB*™ lab details page.

40. When prompted for confirmation, click **OK**.
41. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Systems Operations on AWS
Creating an Elastic Infrastructure
Windows Lab Guide

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Introduction

Overview

In this lab, you use the AWS Management Console and Command Line Interface to build and configure the Elastic Infrastructure necessary to support a web application. This lab is a continuation of the steps you completed in previous labs. In this lab, you will:

1. Create a "gold image" AMI from a 'model' web server for use with auto-scaling.
2. Create an Elastic Load Balancer to distribute traffic between instances of your AMI.
3. Create and configure an auto-scaling group, scaling policies and launch configuration.
4. Create and configure Cloud Watch Alarms

Objectives

This lab is intended to follow modules that introduce Amazon EC2 and VPC. In this lab, you use the knowledge you gained on EC2 and VPC to create an elastic AWS infrastructure.

In general you access the AWS Management Console using your native environment, and you access the Amazon CLI from the remote bastion host.

Note: Replace `<place-holder>` with data from your lab environment.
Note: As you type the provided commands, the default output will be in JSON format. You may find that running these commands from a PowerShell session on Windows (to take advantage of "`ConvertFrom-Json`") or with the aid of `jq` on Linux improves legibility and navigation.
Note: 4 files are generated on the Bastion instance: aws_cli_config, awsstudent.id.rsa, UserData and lab-details.txt. You have previously seen aws_cli_config and awsstudent_id.rsa. The 2 new files are:

- Lab-details.txt contains output for creating your AMI and for auto scaling.
- UserData contains the data needed to create your AMI.

Topics Covered

This lab covers the following topics:

- Amazon EC2 – Amazon Machine Image for deployment
  - Using EC2 instance UserData
  - Using a script with EC2 instance UserData
- Elastic Load Balancing
  - Creating an Elastic Load Balancer
  - Configuring a HealthCheck
- Amazon EC2 – Auto scaling
  - Auto scaling launch configurations
  - Auto scaling groups
  - Auto scaling policies
- CloudWatch
  - CloudWatch alarms for auto scaling

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Connect to the Bastion Host

Since we do not yet have access to our homogeneous management environment – i.e., the Bastion Host –, the next few steps will be done mostly through the AWS Management Console.

Obtain the Bastion Host's Key Pair

This Bastion Host and its associated settings (including its Key Pair) have been created automatically for you when you started the lab. The instructions to download the key file are under a previous section of this guide called *Error! Reference source not found.*, which you should have already completed by now.

Add your current public IP to the Bastion Host's Security Group

It is a security best practice to deny all access to the Bastion Host most of the time. You should only allow access to the Bastion Host from a single IP address – your current public IP address – only during the time you need to interact with it. Once you are done, the recommendation is to revoke all access to the Bastion Host once again.

1. Retrieve your current public IP address by navigating your browser to http://ip.appspot.com
2. Open the **AWS Management Console**.
3. Select the **VPC** service.
4. From the menu on the left side, click **Security Groups**.
5. Click the security group which name contains **BastionHostSecurityGroup**. Its description reads **"Security group for the Bastion Host"**.
6. On the bottom pane, click the **Inbound** tab.
7. From the **Create a new rule dropdown**, select RDP.
8. In the **Source** field, specify **your public IP address followed by /32**. For example: 12.34.56.78/32.
9. Click the **Add rule** button.
10. Click the **Apply Rule Changes** button. Note: you might need to scroll down to see this button.

Obtain the public IP of the Bastion Host

11. Open the **AWS Management console**.
12. Select the **EC2** service.
13. From the menu on the left side, click **Instances**.
14. **Select the BastionHost instance** by clicking on the box next to its name.
15. The Public IP is displayed on the bottom pane. Please write it down.

Retrieve the bastion instance password

Before connecting to the bastion using RDP, you must retrieve the instance's password for the Windows Administrator account.

16. In your browser, switch to the **qwikLAB** tab.
17. In the **Lab Connection** portion of the page, copy the **Password** in the **RDP** section – you may also paste it into a text file for later retrieval.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Note: This is the password you use to log into the bastion instance using the remote desktop client.

18. If the computer you are using to complete this lab is Windows-based, proceed to the next section. If you are using an OS X or Linux-based machine, proceed to the section appropriate to your operating system.

Connecting to the bastion instance (Windows)

Important: Proceed to one of the later sections entitled "Connect to the bastion instance (OS X)" or "Connect to the bastion instance (Linux)" if you are using a Linux-based laptop to perform this exercise. This section is *only* for Microsoft Windows users. There is no need for you to perform the steps in all sections. Only complete the section that matches your operating system.

19. On your local computer click **Start > Run**, type **MSTSC** and click **OK** to start the local RDP client.
20. Type or paste the Elastic IP address of the bastion instance in the **Computer** field.
21. Click **Show Options**, and type **\Administrator** in the **User name** field.

Note: Specifying the user name as "\Administrator" (with a leading backslash) differentiates the bastion Administrator account from the Administrator account on your local computer.

22. Click **Connect**.
23. When prompted, type the RDP **Password** you noted then click **OK**.
24. Click **Yes** if you see a certificate verification message similar to "the identity of the remote computer cannot be verified:"

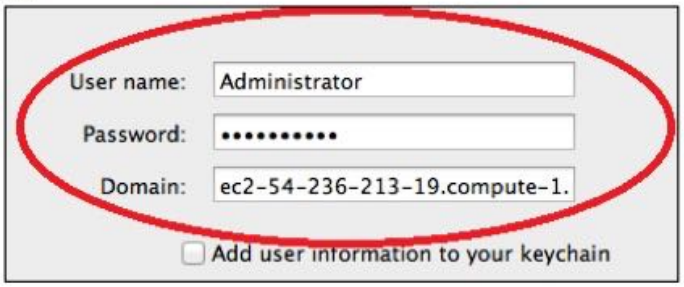
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

25. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (OS X)

Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." This section is for OS X users *only*. If you are using a different Linux-based operating system, please proceed to the section entitled "Connect to the bastion instance (Linux)." There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

26. Open the Remote Desktop Connection for Mac application.
27. Type the bastion instance's IP address you noted in the **Computer** field and click **Connect**.
28. When prompted, in the **User name** field, type Administrator and then type the **Password** you noted from the qwikLAB page. The Domain should be automatically set to the EC2 Instance DNS name, and you can ignore it.



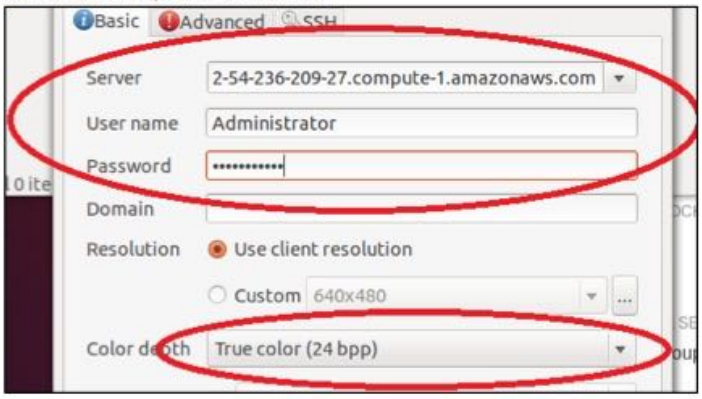
29. Click **OK**.
30. Click **Connect** if you see a verification message similar to "the server name is incorrect."
31. Proceed to the section entitled "Using the AWS CLI."

Connecting to the bastion instance (Linux)

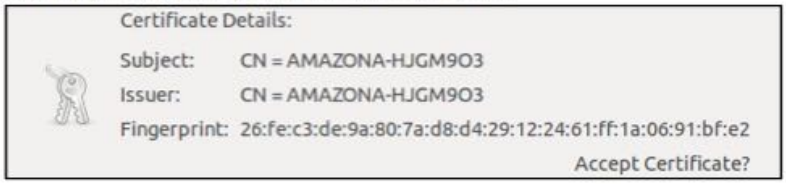
© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Important: If you completed the previous section, you do not need to complete the steps here. Please proceed to the section entitled "Using the AWS CLI." Only complete this section if you are using a non-Macintosh, Linux-based laptop to perform this exercise. This section is *only* for Linux users. There is no need for you to perform the steps in each section. Only complete the section that matches your operating system.

32. Open the Remmina Remote Desktop Client.
33. Type the bastion instance's IP address you noted in the **Server** field.
34. In the **User name** field type **Administrator**, and in the **Password** field, type the password from the qwikLAB page.
35. Optionally, choose a **Color depth** that your bandwidth supports (in this example 'True color (24 bpp)') for a nicer remote desktop environment.



36. Click **Connect**.
37. Click **OK** when prompted to accept the remote certificate.



© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Creating Amazon EC2 Instances and Custom AMIs

Deploying your Amazon AMI with Customizations

An essential piece of an elastic infrastructure is a base AMI, or image, used for auto-scaling configuration management. Use cases for custom AMIs include: keeping operating system and application updates deployed consistently, keeping your code deployed consistently and as a means for replacing failed systems. It is important to note that, should an update cause your application to fail, you have the ability to easily roll-back to a known, stable AMI. Another important use for an AMI, and the one focused on in this lab (and for the remainder of the day), is auto scaling. A custom AMI, along with idempotent customer application code, allows the auto scaling functionality that AWS offers to provide you an elastic infrastructure.

Focus on building a custom AMI from an Amazon AMI. The example you are working with is an IIS application. The application requires the installation of additional deployment packages and administrative tools. The packages include: OS updates, IIS and other application-specific items. You can "bake" all of your prerequisites into an image, as you might be doing now as a traditional operations engineer, and then deploy them in an automated fashion to ensure uniformity.

38. In the next steps, you will create a `Microsoft Windows Server 2012 Base AMI` based instance with user data customizations.
39. First, download a user data file from https://us-west-2-aws-training.s3.amazonaws.com/sys-ops-vpc-elastic-infrastructure-windows-v1.1/static/UserData.txt and save it under **`C:\Users\Administrator\UserData.txt`**. Take care to ensure that there are no special characters while saving the file.
40. In your remote desktop session, open **`C:\Users\Administrator\lab-details.txt`** and record the **AMI ID.** The details available in this file will be used throughout the lab.
41. Create a new Security Group called **ALL** with RDP (3389), HTTP (80) and HTTPS (443) access under **VPC > Security Groups** for your VPC. Provide access to all networks (0.0.0.0/0). Do not forget to click on **Apply Rule Changes** for the changes to take effect. Make a note of the security group id available under **Details > Group ID** as you will use the same in the next step.

42. On the Bastion Host, click the **Windows PowerShell** icon in the taskbar.
43. At a PowerShell prompt, type the following command to create your AMI (substituting your AMI ID in place of the highlighted parameter. Do not use .pem or .ppk extension for the key name):

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
aws ec2 run-instances --key-name <qwiklab-xxx-xxxxx> --instance-type
m1.small --image-id <AMIId> --security-group-ids <ALLSecurityGroupID>
--subnet-id <PublicSubnetID> --user-data
file://C:\Users\Administrator\UserData.txt --associate-public-ip-
address
```

44. The resulting output consists of useful data such as the Instance ID and KeyName. Your output will look similar to the following:

```
...

"Instances": [
...
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2014-01-18T20:22:12.000Z",
            "PrivateIpAddress": "10.50.2.95",
            "ProductCodes": [],
            "VpcId": "vpc-a311f5c6",
            "StateTransitionReason": null,
            "InstanceId": "i-e1ec54c0",
            "ImageId": "ami-05355a6c",
            "PrivateDnsName": "ip-10-50-2-95.ec2.internal",
            "KeyName": "bastion-qlstack2-lab-135-1390075369",
            "SecurityGroups": [
                {
                    "GroupName": "qlstack2-lab-135-1390075369-
WebSecurityGroup-1UD5166X1HSYY",
                    "GroupId": "sg-5aa3663f"
                }
            ],
            "ClientToken": null,
            "SubnetId": "subnet-92b8e2d4",
            "InstanceType": "m1.small",
...
                "AvailabilityZone": "us-east-1a"
            },
...
```

45. On the management console find your way to **EC2 > Instances** and name the newly created instance as **AMI Instance** by clicking the **Name** column on the instance.

Converting Instances into Custom AMIs.

Having deployed an instance from the Amazon AMI with your customizations in the UserData.txt file, you will now create a custom AMI to use for deployment of instances with Auto Scaling.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

46. Gather your Instance id from the output of the previous command or on the console: **EC2 > Instances** and note the Instance ID from the center top panel.
47. Check the instance creation state by replacing retrieved data from the previous output with the <> section.

```
aws ec2 describe-instance-status --instance-id <Instance-Id>
```

Note: You will not see any indication that bootstrapping is complete. The 'Running' state is equivalent to an 'instance is powered on' indicator. You could, at this point, connect to the newly created and look at the logs to see if bootstrapping is complete. Decrypt the instance password using the qwikLAB .pem file. This step could take a while. Bootstrapping in this lab, would have created a file called 'bootstrap.log' on the desktop. You could read the file to find out if bootstrapping completed successfully. Make sure you only proceed to the next step after you confirm that bootstrapping has completed successfully.
The output should be similar as below.

```
**********************
Windows PowerShell transcript start
Start time: 20140212204502
Username  : WIN-M2DE6S1LRJU\Administrator
Machine   : WIN-M2DE6S1LRJU (Microsoft Windows NT 6.2.9200.0)
**********************
Transcript started, output file is
c:\users\Administrator\Desktop\bootstrap.log
[Start of script C:\Program
Files\Amazon\Ec2ConfigService\Scripts\UserScript.ps1]
[Installing IIS]
Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             Success        {Common HTTP Features, Default
Document, D...
WARNING: Windows automatic updating is not enabled. To ensure that your
newly-installed role or feature is
automatically updated, turn on Windows Update.
[Preparing to download files]
[Downloading https://s3.amazonaws.com/xxxxx/xxxxxxxxxxxxx/cpuload.vbs]
[Downloading
https://s3.amazonaws.com/xxxxx/xxxxxxxxxxxxx/examplefiles-elb.zip]
[Using .NET to expand zip file
c:\users\Administrator\Desktop\examplefiles-elb.zip]
[Expansion complete for c:\users\Administrator\Desktop\examplefiles-
elb.zip]
[Downloads complete]
[Bootstrap completed successfully]
**********************
Windows PowerShell transcript end
End time: 20140212204654
**********************
```

48. You can also test if the userdata was executed successfully by verifying if an IIS based web page is accessible using the instance IP address on a browser.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

49. You need to run Sysprep before shutting down the instance. Click on **Start > EC2ConfigService settings > Image > Random (Retrieve password from console; set at next service start)**.
50. Click on **Shutdown with Sysprep > Yes**. Observe the instance execute Sysprep and shutdown. Be patient as this step could take a few minutes.
51. When the instance is stopped, type the following command to create an AMI from it (replacing the highlighted parameter with your Instance Id):

```
aws ec2 create-image --name WEBSERVER --instance-id <Instance-Id>
```

Broken down, this command calls `aws ec2` service. It then calls the `create-image` operation. And finally, it calls the parameters `instance-id` and `name`, which tags the image with a name. Your output will be similar to the following, indicating a successful custom AMI creation, and take note of this ID:

```
"ImageId": "ami-a97c37c0"
```

52. Validate your AMI creation in the EC2 Dashboard by clicking **Images > AMIs**. The status should change from "pending" to "available." You can proceed to the next step while your AMI is being created.

Amazon EC2 Auto Scaling

Understanding and Employing Auto Scaling

This section of the lab will have you build the Auto Scaling functionality needed to scale up and down the lab web application that will be explored in subsequent labs. Auto Scaling involves several steps, creating a Lunch Configuration, an Auto Scaling group, and one or more Auto Scaling policies. Additionally, Auto Scaling compliments Elastic Load Balancer to distribute traffic as the load increases or decreases.

You can choose to create an ELB and configure your Auto Scaling Group either by using the management console or by using aws cli. In the below section, we will use the management console. If you would like to do the same in CLI, check the Bonus Activity section that is available at the end of this lab.

Creating an Elastic Load Balancer

ELB, or Elastic Load Balancer works well with the Auto Scaling functionality. ELB obfuscates the servers behind it so that the general web user will not need to know the name of the server they are actually connected to, rather, they'll connect to the ELB. The ELB does not change throughout the scaling process. Here, you will create an ELB for your web servers to auto scale

53. On the management console select **EC2 > Load Balancers > Create Load Balancer**
54. In the **Load Balancer Name** field type **ASLoadBalancer**
55. In the **Create LB inside** field choose your **VPC** and click on **Continue**
56. As displayed in the picture below, enter the following.
    g. **Ping Port** field enter **80.**
    h. **Ping Path** filed enter **/index.htm**.
    i. **Response Timeout** field enter **5.**
    j. **Health Check Interval** field enter **60 seconds**.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

k. **Unhealthy Threshold** field enter **4**.
l. **Healthy Threshold** field choose **2** and click **Continue**.

57. On the next screen select the two Public Subnets by choosing and click **Continue.**
58. Next choose the **ELBSecurityGroup** and click **Continue.**
59. Make sure you do not select any instance on the next screen as Auto Scaling will add instances behind the ELB later. Click on **Continue**.
60. Finally click on **Create** and click on **Close**.
61. After the ELB is created, you can select the ELB and view its properties.

Creating an Auto Scaling Launch Configuration and Auto Scaling Group

A launch configuration is primarily used to apply specific modifications to instances as they are created programmatically via the Auto Scale up policies. It will consist of configuration that is used to create EC2 instances. That configuration is similar to what you input when creating your instance earlier in this lab.

62. On the management console choose **EC2 > AUTO SCALING > Launch Configurations**.
63. Click on **Create Auto Scaling group**.
64. Select **Create launch configuration**.
65. On the **Create Launch Configuration** screen choose **My AMIs** and **Select** the AMI you created. Select **General purpose** on the left of the screen. Choose **m1.small** on the center of the screen. Select Next**: Configure details**.
66. On the next screen under **Name** enter **ASLaunchConfiguration** and click on **Next: Add Storage**.
67. Click on **Next: Configure Security Group.**
68. Choose **Select an existing security group** and select the **WebSecurityGroup** from the list and click on **Review**.
69. Click on **Create launch configuration**.
70. On the Select an existing key pair screen, select **Choose an existing key pair** and select the qwikLAB key pair. Choose I acknowledge and click on **Create launch configuration.**
71. Next you will create an Auto Scaling group. This portion will help determine the minimum and maximum number of instances among other details. Do not forget to replace relevant text with values from the lab-details.txt file.
72. On the **Create Auto Scaling Group** screen, enter the following details **Group name** as **WebserversASGroup, Group size** as **2**. **Network** as **VPC** and **Subnet** select both the public subnets. Expand the **Advanced Details** section and select **Receive traffic from Elastic Load Balancer(s)** check box. Further select your **ELB (ASLoadBalancer).** Under **Health Check Type** select **ELB**. Enter **300** under the **Health Check Grace Period** field and select **Next: Configure scaling policies**.
73. On the next screen choose **Use scaling policies to adjust the capacity of this group** and set **Scale between 2** and **4**. This value defines the minimum and maximum number of instances that would be a part of your Auto Scaling group. This group is configured as a 2-4-2 configuration, that is, `minsize-maxsize-desiredcapacity`. In this configuration, a minimum of two instances will be kept up and replaced if health checks fail. Auto Scaling will increase to 4 instances as load increases, and then reduce back down to the desired size of 2 when load decreases.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Configure Auto Scale Policies and Create Alarms

Auto Scale policies configure Auto Scaling to automatically scale based on conditions. Use cases include, CPU, memory or disk thresholds. You will configure both scale up and scale down policies.

74. Configure the scale up policy under the **Increase Group Size** section. Enter a name of your choice.
75. Under the **Execute policy when** section select **Add new alarm**.
76. On the **Create Alarm** screen, uncheck **Send a notification** check box. You can however choose to be notified by creating a topic. Make sure you select Average CPU Utilization to be **>= 50** percent. **For at least:** and **consecutive period(s)** fields must be set to **1 and 1 Minute**.
77. Enter **ASScaleUp** under **Name of Alarm** and click on **Create Alarm**.
78. **Take the action:** should be set to **Add 1** instance. **Add then wait** that refers to a cool down period must be **300**.
79. Similarly under the **Decrease Group Size** section create a scale down policy. Enter a name of your choice.
80. Under the **Execute policy when** section select **Add new alarm**.
81. On the **Create Alarm** screen, uncheck **Send a notification** check box. Like above you can choose to be notified by creating a topic in production. Select Average CPU Utilization to be **<= 35** percent. **For at least:** and **consecutive period(s)** fields must be set to **1 and 1 Minute.**
82. Enter **ASScaleDown** under **Name of Alarm** and click on **Create Alarm**
83. **Take the action:** should be set to **Remove 1** instance. **Add then wait** to **300**.
84. Click on **Next: Configure Notifications**.
85. You can add a notification to be alerted about any change in the Auto Scaling Group.
86. Click on **Review** and verify if you have selected the right options and then click on **Create Auto Scaling group**.
87. Once you receive the **Successfully created Auto Scaling group** message click the **Close** button.
88. You can now select your newly created Auto Scaling group and review its settings.
89. On the management console choose **EC2 > Instances** and verify that two instances are being created.
90. Find your way to **EC2 > Load Balancers** and click the load balancer you created. Now select the **Instances** tab and look for the two instances that were created by Auto Scaling. Keep refreshing the screen until both the instances are **In Service**. This indicates that the instances are ready to accept web traffic.
91. Under the **Description** tab you should now be able to find a **DNS Name:** `http://asloadbalancer-xxxxxxxxx.xx-xxxx-x.elb.amazonaws.com`. Copy the same and enter it in the address bar of your browser and you should see a web page that is served from one of the backend instances.
92. You can add some CPU load on one of the backend instance to see Auto Scaling scale up. From the Bastion host log into one of the backend instances created by Auto Scaling via RDP. On the desktop you should find a file named 'cpuload.vbs'.
93. **Note:** Do not execute the file by double clicking it. Rather right click on it and select 'Open with Command Prompt'. This step would add load on the CPU. You can execute another cpuload.vbs load, in case the current file does not add sufficient load on the instance.
94. As per the Auto Scaling policy to scale up, you should now see a new instance being created.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Bonus Activity

Note: Please proceed with the below steps only if you skipped the ELB and Auto Scaling management console (GUI) steps.

Creating an Elastic Load Balancer using aws cli

ELB, or Elastic Load Balancer is required for Auto Scaling functionality. ELB obfuscates the servers behind it so that the general web user will not need to know the name of the server they are actually connected to, rather, they'll connect to the ELB. The ELB does not change throughout the scaling process. Here, you will create your ELB for your web servers to reside and Auto Scale behind.

1. Close the PowerShell prompt if open and open the Command Prompt.

   Note: All the below commands must be executed from the Command Prompt only.

2. The names of your resources you will need can be found on your Bastion Host at `C:\Users\Administrator\lab-details.txt`

```
aws elb create-load-balancer --load-balancer-name ASLoadBalancer --
listeners
"LoadBalancerPort=80,InstancePort=80,Protocol=http,InstanceProtocol=http"
```

   The resulting output will be similar to the following:

```
"DNSName": "ASLoadBalancer-173888080.us-east-1.elb.amazonaws.com"
```

3. You will need to configure the health checks:

```
aws elb configure-health-check --load-balancer-name ASLoadBalancer --
health-check
Target=HTTP:80/index.htm,Interval=60,Timeout=5,UnhealthyThreshold=4,Health
yThreshold=2
```

   Your output will be similar to this:

```
          "Health Check": {
              "HealthyThreshold": 2,
              "Interval": 15,
              "Target": "HTTP:80/index.htm",
              "Timeout": 5,
              "UnhealthyThreshold": 3
```

4. Validate your ELB with the following:

```
aws elb describe-load-balancers
```

   The results should be similar to the following:

```
{
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
    "LoadBalancerDescriptions": [
        {
            "Subnets": [
                "subnet-3bcd8a13",
                "subnet-6dd2e819"
            ],
            "CanonicalHostedZoneNameID": "Z3DZXE0Q79N41H",
            "CanonicalHostedZoneName": "ASLoadBalancer-173888080.us-
east-1.elb.amazonaws.com",
            "ListenerDescriptions": [
                {
                    "Listener": {
                        "InstancePort": 80,
                        "LoadBalancerPort": 80,
                        "Protocol": "HTTP",
                        "InstanceProtocol": "HTTP"
                    },
                    "PolicyNames": []
                }
            ],
            "Health Check": {
                "HealthyThreshold": 2,
                "Interval": 15,
                "Target": "HTTP:80/index.htm",
                "Timeout": 5,
                "UnhealthyThreshold": 3
            },
            "VPCId": "vpc-7b89691e",
            "BackendServerDescriptions": [],
            "Instances": [],
            "DNSName": "ASLoadBalancer-173888080.us-east-
1.elb.amazonaws.com",
            "SecurityGroups": [
                "sg-269a5b43"
            ],
            "Policies": {
                "LBCookieStickinessPolicies": [],
                "AppCookieStickinessPolicies": [],
                "OtherPolicies": []
            },
            "LoadBalancerName": "ASLoadBalancer",
            "CreatedTime": "2014-01-27T03:47:28.350Z",
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b"
            ],
            "Scheme": "internet-facing",
            "SourceSecurityGroup": {
                "OwnerAlias": "972013970906",
                "GroupName": "qlstack2-lab-135-1390793521-
ELBSecurityGroup-Q95O1B9Y5F5H"
            }
        }
    ]
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
}
```

Creating an Auto Scaling Launch Configuration and Auto Scaling Group using aws cli

A launch configuration is primarily used to apply specific modifications to instances as they are created programmatically via the Auto Scale up policies. They will consist defaults data that gets put to EC2. That data is similar to what you input when creating your instance earlier in this lab.

5. Input the following to create an Auto Scaling launch configuration, replacing the appropriate fields with your own data. **Note:** the `customAMIid` is the AMI id of the AMI you created.

```
aws autoscaling create-launch-configuration --instance-type m1.small
--key-name <qwiklab-xxxx-xxxxx> --launch-configuration-name
ASLaunchConfiguration --image-id <customAMIid> --security-groups
<WebSecurityGroup>
```

   With this command `aws autoscaling`, you are enacting the EC2 Auto Scaling service with the operation `create-launch-configuration` and the following parameters:
   Image-id – This ID is the ID associated with your custom AMI that you created earlier in the lab.
   Instance-type – This is your instance "size", IE: t1.micro, m1.small.
   Key – This is the key pair to use.
   Launch-configuration-name – This name will be used to refer to the launch configuration.
   Security-groups – This can either be the name or the Group ID.
   The output will be an empty set, again.

6. Then, double check the launch configuration parameters to verify the command completed with expected results by typing:

```
aws autoscaling describe-launch-configurations
```

   This command will display similar resulting information:

```
{
    "LaunchConfigurations": [
        {
            "UserData": null,
            "EbsOptimized": false,
            "LaunchConfigurationARN": "arn:aws:autoscaling:us-east-
1:972013970906:launchConfiguration:f04a67b2-d282-405f-b0e2-
4112a4f0fa14:launchConfigurationName/ASLaunchConfiguration",
            "InstanceMonitoring": {
                "Enabled": true
            },
            "ImageId": "ami-05355a6c",
            "CreatedTime": "2014-01-27T04:18:13.156Z",
            "BlockDeviceMappings": [],
            "KeyName": "qlstack2-lab-135-1390793521",
            "SecurityGroups": [
                "sg-2f9a5b4a"
            ],
            "LaunchConfigurationName": "ASLaunchConfiguration",
            "KernelId": null,
            "RamdiskId": null,
            "InstanceType": "m1.small"
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
        }
    ]
}
```

Note that you could also inject userdata at this point, in the Auto Scaling launch configuration.

7. Next you will create an Auto Scaling group to determine the resources to be used for the Auto Scaling of the web instances. Again, replace <> text with your values

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name
WebserversASGroup --launch-configuration-name ASLaunchConfiguration --min-
size 2 --max-size 4 --desired-capacity 2 --health-check-type ELB --health-
check-grace-period 300 --load-balancer-name ASLoadBalancer --vpc-zone-
identifier <PublicSubnet1>,<PublicSubnet2>
```

This group is configured as a 2-4-2 configuration, that is, `minsize-maxsize-desiredcapacity`. In this configuration, a minimum of two instances will be kept up and replaced if health checks fail. Auto Scaling will increase to 4 instances as load increases, and then reduce back down to the desired size of 2 when load decreases. The `launch-configuration-name` refers back to the launch configuration you created.

8. Validate your new Auto Scaling group with the following:

```
aws autoscaling describe-auto-scaling-groups
```

The result will be similar to the following:

```
{
    "AutoScalingGroups": [
        {
            "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-
1:972013970906:autoScalingGroup:3ec12c00-8407-4a2a-8504-
c104fa1a8170:autoScalingGroupName/WebserversASGroup",
            "HealthCheckGracePeriod": 30,
            "SuspendedProcesses": [],
            "DesiredCapacity": 2,
            "Tags": [],
            "EnabledMetrics": [],
            "LoadBalancerNames": [
                "ASLoadBalancer"
            ],
            "AutoScalingGroupName": "WebserversASGroup",
            "DefaultCooldown": 300,
            "MinSize": 2,
            "Instances": [
                {
                    "InstanceId": "i-1b5f173b",
                    "AvailabilityZone": "us-east-1b",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService",
                    "LaunchConfigurationName":
"ASLaunchConfiguration"
                },
                {
                    "InstanceId": "i-185f1738",
                    "AvailabilityZone": "us-east-1b",
                    "HealthStatus": "Healthy",
                    "LifecycleState": "InService",
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
                    "LaunchConfigurationName":
"ASLaunchConfiguration"
                }
            ],
            "MaxSize": 4,
            "VPCZoneIdentifier": "subnet-3acd8a12",
            "TerminationPolicies": [
                "Default"
            ],
            "LaunchConfigurationName": "ASLaunchConfiguration",
            "CreatedTime": "2014-01-27T05:12:58.174Z",
            "AvailabilityZones": [
                "us-east-1b"
            ],
            "HealthCheckType": "ELB"
        }
    ]
}
```

9. Next, you will validate the Auto Scaling instance that is recognized within the Auto Scaling group.

```
aws autoscaling describe-auto-scaling-instances
```

Your output should be similar to:

```
{
    "AutoScalingInstances": [
        {
            "AvailabilityZone": "us-east-1c",
            "InstanceId": "i-78018c1c",
            "AutoScalingGroupName": "WebserversASGroup",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService",
            "LaunchConfigurationName": "ASLaunchConfiguration"
        },
        {
            "AvailabilityZone": "us-east-1c",
            "InstanceId": "i-7a018c1e",
            "AutoScalingGroupName": "WebserversASGroup",
            "HealthStatus": "HEALTHY",
            "LifecycleState": "InService",
            "LaunchConfigurationName": "ASLaunchConfiguration"
        }
    ]
}
```

10. Validate Auto Scaling functionality by terminating an instance (not the BASTION or NAT instance!). A new instance should be recreated. Gather an Instance ID from the previous output, then type.

```
aws ec2 terminate-instances --instance-ids <instance-id>
```

The output should be similar to the following:

```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-7a018c1e",
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

```
                "CurrentState": {
                    "Code": 48,
                    "Name": "terminated"
                },
                "PreviousState": {
                    "Code": 48,
                    "Name": "running"
                }
            }
        ]
    }
```

11. Now, describe the Auto Scaling instances again:

```
aws autoscaling describe-auto-scaling-instances
```

And view that 2 instances are again available. If you do not see 2 instances, wait for a minute or two for the instance to finish its boot process and try again. You could also validate your instances from the management console. Go to **EC2 > Instances > Instances** and validate 2 m1.small instances are now being launched or have been launched.

Configure Auto Scale Policies using aws cli

Auto Scale policies configure Auto Scaling to happen automatically based on conditions. Use cases include, CPU thresholds, memory thresholds and disk thresholds. You will configure both scale up and scale down policies.

12. Return to the terminal window on your Bastion Host to perform these steps.
13. Configure the scale up policy first

```
aws autoscaling put-scaling-policy --auto-scaling-group-name
WebserversASGroup --policy-name ASscaleUPpolicy --scaling-adjustment
1 --adjustment-type ChangeInCapacity --cooldown 300
```

Results will be an ARN:

```
{
    "PolicyARN": "arn:aws:autoscaling:us-east-
1:972013970906:scalingPolicy:a285e3e1-cbdd-40ca-88d7-
d6f97d2d8824:autoScalingGroupName/WebserversASGroup:policyName/ASsca
leUPpolicy"
   }
```

14. Now, create the scale down policy.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name
WebserversASGroup --policy-name ASscaleDownpolicy --scaling-
adjustment -1 --adjustment-type ChangeInCapacity --cooldown 300
```

Again, the results will be an ARN:

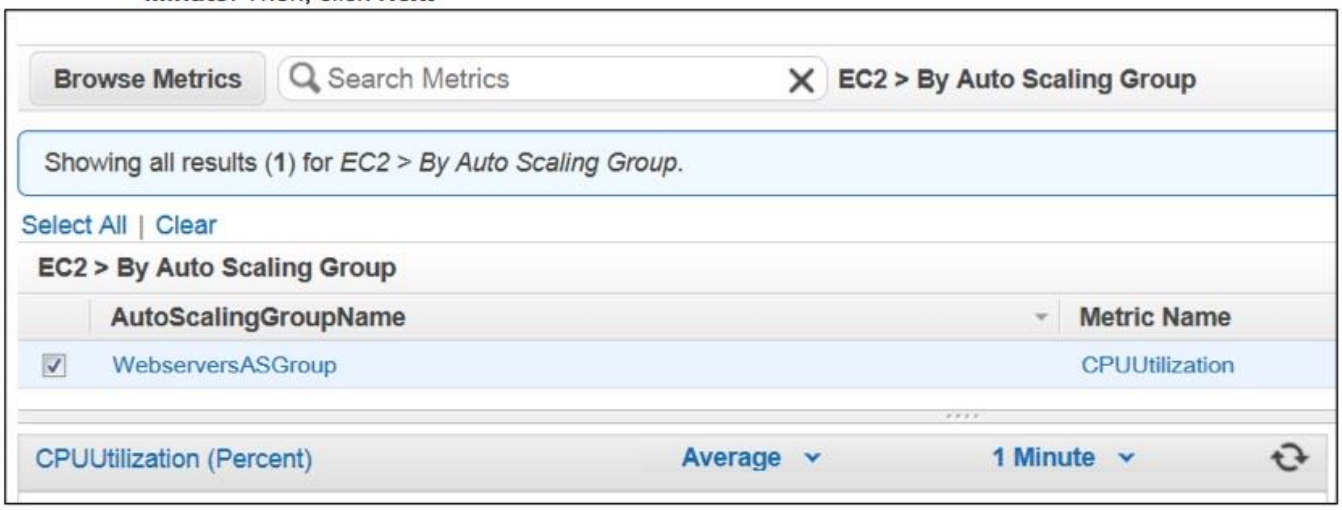
```
{
    "PolicyARN": "arn:aws:autoscaling:us-east-
1:972013970906:scalingPolicy:df5f7b69-bc80-4689-b1ab-
ae5d485c9432:autoScalingGroupName/WebserversASGroup:policyName/ASsca
leDownpolicy"
   }
```


© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

Configure Cloud Watch CPU Alerts

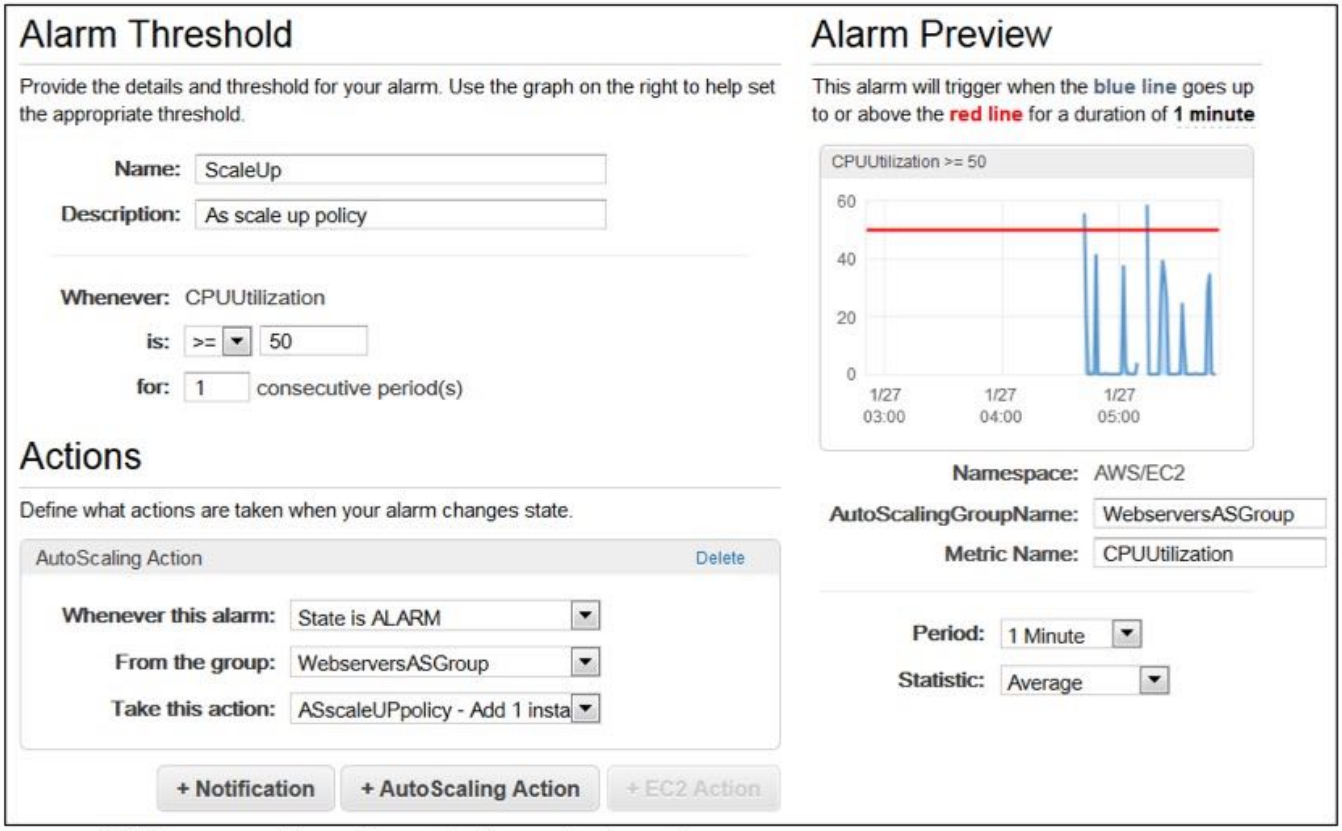
You need to now configure your Cloud Watch alarm to alert Auto Scaling when aggregate CPU is above and when it is below the threshold you set to perform scale up and scale down operations. It takes a few minutes for Cloud Watch to begin gathering statistics and to recognize your new Auto Scaling group.

15. First, create the scale up Cloud Watch alarm.
16. Return to the management console and go to **Services > All AWS Services > Cloud Watch**
17. Click **Alarms**, and then click **Create Alarm**.
    Search for **By Auto Scaling Group:** and chose the alarm associated with "**WebserversASGroup**" and "**CPUUtilization**", and change the Period from 5 Minutes of **1 Minute**. Then, click **Next**



18. Provide a meaningful scale up name and description, and set the utilization to **>= 50**, for **1** minute. Under **Actions > delete** the Notification. Click on **+Auto Scaling Action** > **From the group:** select **WebserversASGroup** and under **Take this action:** select **ASscaleUPpolicy** and click on **Create Alarm.**

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.



19. Now, you will need to create the scale down alarm
20. Click **Alarms**, and then click **Create Alarm**.
21. Search for **By Auto Scaling Group:** and chose the alarm associated with "**WebserversASGroup**" and "**CPUUtilization**", and change the Period from 5 Minutes of **1 Minute**. Then, click **Next**.
22. Provide a meaningful scale down name and description, and set the utilization to **<= 35**, for **1** minute. Under **Actions > delete** the Notification. Click on **+Auto Scaling Action** > **From the group:** select **WebserversASGroup** and under **Take this action:** select **ASscaleDownpolicy** and click on **Create Alarm**.
23. View your newly created Metrics and history by selecting one of your policies at the **Alarms** screen:

24. Find your way to **EC2 > Load Balancers** and click the load balancer you created. Now select the **Instances** tab and look for the two instances that were created by Auto Scaling. Keep refreshing the screen until both the instances are **In Service**. This indicates that the instances are ready to accept web traffic.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

25. Under the **Description** tab you should now be able to find a **DNS Name:** `http://asloadbalancer-xxxxxxxxx.xx-xxxx-x.elb.amazonaws.com`. Copy the same and enter it in the address bar of your browser and you should see a web page that is served from one of the backend instances.
26. You can add some CPU load on one of the backend instance to see Auto Scaling scale up. From the Bastion host log into one of the backend instances created by Auto Scaling via RDP. On the desktop you should find a file named 'cpuload.vbs'.
    Note: Do not execute the file by double clicking it. Rather right click on it and select 'Open with Command Prompt'. This step would add load on the CPU. You can execute another cpuload.vbs load, in case the current file does not add sufficient load on the instance.
27. If you notice that instances are getting created and then terminated it could be because the load on the unused instance is very low. So log into the instance from the Bastion Host and execute one or more 'cpuload.vbs' scripts.
28. As per the Auto Scaling policy to scale up, you should now see a new instance being created.
29. If you notice that instances are getting created and then terminated it could be because the load on the unused instance is very low. So just log into the instance from the Bastion Host and execute one or more 'cpuload.vbs' scripts.
30. On the qwikLAB screen click on **END LAB** to end the lab exercise.

Conclusion

Congratulations! You now have successfully:

- Bootstrapped an instance from a file
- Repackaging for conversion to an AMI
- Configured an Auto Scaling Launch Configuration
- Configured Auto Scaling Groups
- Configured Auto Scaling Policies
- Configured an Elastic Load Balancer

Configured Cloud Watch Alarms for Auto Scaling activities

End Your Lab

31. To log out of the AWS Management Console, from the menu, click **awsstudent @ [YourAccountNumber]** and choose **Sign out** (where [YourAccountNumber] is the AWS account generated by *qwikLAB*™).

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.

32. Close any active SSH client sessions or remote desktop sessions.
33. Click the **End Lab** button on the *qwikLAB*™ lab details page.

34. When prompted for confirmation, click **OK**.
35. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.

Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

© 2013, 2014 Amazon Web Services, Inc. and its affiliates. All rights reserved.