Lab 1: Control resource consumption using tagging strategies v1.0.4

Sunday, August 9, 2020 7:03 PM

**

requirements and instance type standardization. By deploying this solution, your organization will have account wide standardization of resources, as well as meaningful metadata attached to every instance.

Objectives

After completing this lab, you will be able to:

- Configure rules in AWS Config to identify non-compliant resources in your environment, including:
    - Enforcing the use of standardized tags.
    - Enforcing the deployment using approved instance types.
- Auto-remediate the non-compliant resources using AWS Config.
- Prevent creation of non-compliant resources based on required tags in IAM policies.

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, Mac OS X, or Linux (Ubuntu, SuSE, or Red Hat).
- The qwikLABS lab environment is not accessible using an iPad or tablet device, but you can use these devices to access the student guide.
- For Microsoft Windows users: Administrator access to the computer.
- An Internet browser such as Chrome, Firefox, or IE9 (previous versions of Internet Explorer are not supported).

Duration

This lab will require **60** minutes to complete.

Start Lab

1. At the top of your screen, launch your lab by choosing **Start Lab**

   This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

   ℹ If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing **Open Console**

   This opens an AWS Management Console sign-in page.

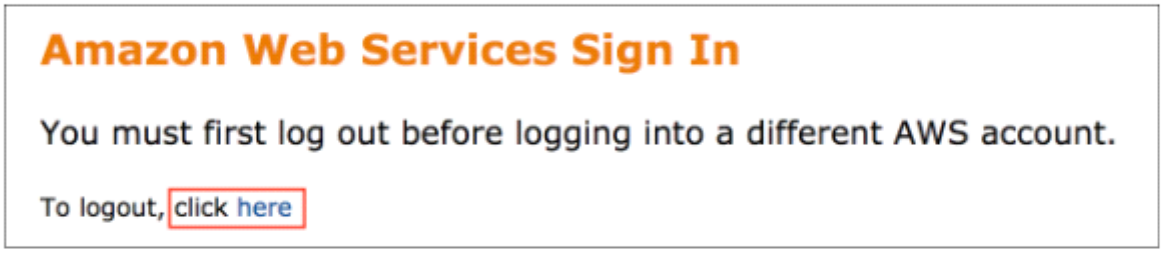
3. On the sign-in page, configure:

   - **IAM user name:** `awsstudent`
   - **Password:** Paste the value of **Password** from the left side of the lab page
   - Choose **Sign In**

⚠ **Do not change the Region unless instructed.**

Common Login Errors

Error: You must first log out



If you see the message, **You must first log out before logging into a different AWS account:**

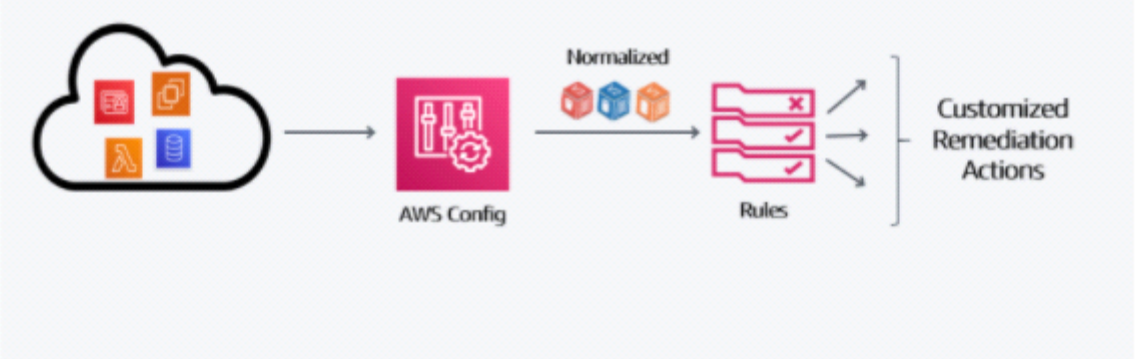
- Choose **click here**
- Close your browser tab to return to your initial lab window
- Choose Open Console again

Introduction

Imagine you are the Solution Architect for Example Corp. When the lease was expiring on the old on-premises data center, there was a rush to lift and shift servers to AWS. Now that the migration is complete, you can start to re-architect with cost in mind.

You noticed that resources deployed in AWS were not being tagged consistently using Example Corp's standard tags. This made it difficult to associate resources and their costs with each department, application, and environment that is running in AWS. In addition, you also observed that some applications were not using approved instance types in accordance with Example Corp's published architecture standards. This can drive unnecessary costs due to over-provisioned resources in non-production environments.

In this lab, you will use AWS Config to enforce tagging and instance type standardization. Next, you will apply an IAM policy to your Developers user group to prevent users from launching EC2 instances without populating the appropriate tags. Finally, once configured, you will test each of these items to confirm that the controls implemented successfully.



Task 1: Enable AWS Config

Before you can start monitoring your AWS resources for compliance, you must enable AWS Config in your environment. AWS Config provides a detailed view of the configuration of AWS resources in your account. Additionally, you can observe how resources relate to one another, past configurations, and how the relationships change over time.

Task 1.1: Set up AWS Config

4. Open the **AWS Config Console** by selecting **Services** and typing `config` in the filter box.

5. Choose **Config**.

6. Choose **Get started**

7. In the **Resource category** section, Select **Record specific resource types**.

8. Within **Resource Type** select **AWS EC2 Instance**

9. In the **AWS Config role** section, select the option **Choose a role from your**

9. In the **AWS Config role** section, select the option **Choose a role from your account**.

10. For **Role name** select **LabConfigServiceRole**.

11. Choose **Next**

12. No changes on this page, choose **Next**

13. Choose **Confirm**

    NOTE: A warning message may appear at the top of the page requesting that you update the IAM Policy used by AWS Config. This warning can be ignored for purposes of this lab.

Task 2: Create AWS Config Rules

Now that you have enabled AWS Config, you will establish rules that will check resource compliance with your organization's published architecture standards.

NOTE: You may see a note at the top of the screen that a redesigned AWS Config console is available for use. For this lab, please do not use the redesigned console.

Task 2.1: Add AWS Config Rule to enforce required tags

First, you will create a rule that confirms required tags populated with appropriate values on each of your EC2 instances. Consistent tagging is a

useful tool in managing costs associated with running different applications or workloads in your environment. It can also be helpful in allocating costs to different departments that deploy and operate resources in AWS.

14. On the left hand navigation pane, choose **Rules**
15. Choose **Add rule**
16. Select **Add AWS managed rule** for Rule Type
17. In **AWS Managed Rules** section, enter `required-tags` into the search box.
18. Choose the **required-tags** rule and choose **Next**
19. Configure the following settings on the next screen, leaving the other settings as the default values/selections:

    - For **Name** enter `RequiredTagsCompliance`
    - Remove all values from **Resources** field except for **EC2: Instance**.
    - For **Parameters** enter the following:

      NOTE: Clear the pre-populated value in the **tag1Key** field.

Key	Value
tag1Key	`Department`
tag1Value	`Finance`
tag2Key	`Application`
tag2Value	`Accounts Payable`
tag3Key	`Environment`
tag3Value	`Development,Test,Production`

NOTE: Delete unused **tagKey** and **tagValues** by choosing on Remove.

20. Choose **Next**.

21. In Review and create page, choose **Add rule** to save the rule.

    In the Rules dashboard, you will see the new **RequiredTagsCompliance** rule has a Compliance status of Evaluating. When a new rule is created, existing resources are evaluated for compliance with the defined rule (refresh your page to see the compliance evaluation). Move on to the next step; you will review compliance with this rule later in the lab.

Task 2.2: Add AWS Config Rule to enforce approved Instance Types

The second rule you will establish ensures that your EC2 instances are using an approved instance type. Make sure the selected instance type meets the performance requirements of the supported workload. In addition, that it fits the budget approved to deploy and operate resources in AWS.

22. In the left navigation pane under **AWSConfig**, select **Rules**.

23. Choose **Add rule**

24. Select **Add AWS managed rule** for Rule Type

25. Enter `desired-instance-type` into the search box.

26. Choose the **desired-instance-type** rule and select **Next**

27. Configure the following settings on the next screen, leaving the other settings as the default values/selections:

    - For **Name** enter `ProductionInstanceType`

- For **Scope of changes** select **Tags**.
- For **Tag key** enter `Environment`
- For **Tag value** enter `Production`
- For **Value** field in the **Parameters** section, enter `t3.large, t3.xlarge, t3.2xlarge`

  This setting indicates that one of the specified instance types, should be used for the instances deployed in your environment that have an Environment tag value of Production, to avoid unnecessary costs associated using unapproved instance types.

28. Choose **Next**.
29. In Review and create page, choose **Add rule** to save the rule.

    Instead of just identifying the non-compliant resources, you want to stop them from running. This prevents them from incurring costs in your environment. To do this, you are going to implement an automatic remediation to stop those instances from running so that they resize appropriately.

30. In the left navigation pane under **AWSConfig**, select **Rules**.
31. Select **desired-instance-type** rule.
32. Choose **Add rule**

    - Select **Manage remediation**.
    - For **Select remediation method**, select `Automatic remediation`.
    - For **Remediation action details**, enter `AWS-StopEC2Instance` in the search box.

- Select **AWS-StopEC2Instance**.
- For **Resource ID parameter** select **InstanceId**.
- Copy the **Lab Automation Role ARN** value from the navigation panel to the left of these instructions and paste it in the **Value** field next to **AutomationAssumeRole**.

33. Choose **Save changes**

In the Rules dashboard, you will see the new **ProductionInstanceType** rule has a Compliance status of Evaluating. It sometimes takes a while for AWS Config to evaluate the status of the resources in your environment. In the next task, you will configure a preventative control for launching new EC2 instances to allow Config to complete the initial evaluation.

Task 3: Enable Preventative Controls for Compliance

Enabling AWS Config will allow you to detect and remediate non-compliant resources in a reactive manner. Now, you will use AWS Identity and Access Management to put pro-active controls in place that prevent users from deploying non-compliant resources. With IAM, you can grant different permissions, to different people, for different resources. You will use the capabilities of IAM to ensure that your AP Developers can launch new EC2 instances in the Development environment as needed, but only if they select the appropriate instance type, and configure the required tags at launch to be compliant with your organization's standards.

Task 3.1: Review AWS IAM User Group and Policy

To efficiently manage multiple users that perform the same job duties as one, IAM Groups are used. They collectively administer permissions for all users that are members of that group.

34. Open the **IAM Console** by selecting **Services** and typing `iam` in the filter box.

35. Choose **IAM**.

36. In the left menu, select **Groups**.

    A user group for AP Developers was created allowing users in that group to deploy EC2 instances that meet certain requirements. This allows you to enable self-service capabilities while maintaining compliance with your documented standards.

37. Choose the **APDevelopers** group name.

38. Select the **Permissions** tab.

    This user group has an Inline Policy that grants permissions to the users in the group.

39. Choose the **Show Policy** link next to **CompliantEC2Creation**.

    The inline policy will looks similar to the example below.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*",
      "Effect": "Allow",
```

```
      "Sid": "AllowToDescribeAll"
    },
    {
      "Action": [
        "ec2:RunInstances",
        "ec2:CreateVolume"
      ],
      "Resource": [
        "arn:aws:ec2:*::image/*",
        "arn:aws:ec2:*::snapshot/*",
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*",
        "arn:aws:ec2:*:*:key-pair/*",
        "arn:aws:ec2:*:*:volume/*"
      ],
      "Effect": "Allow",
      "Sid": "AllowRunInstances"
    },
    {
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Application": "Accounts Payable",
          "aws:RequestTag/Department": "Finance",
          "aws:RequestTag/Environment": "Development",
          "ec2:InstanceType": "t3.medium"
        }
      },
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Effect": "Allow",
      "Sid": "AllowRunInstancesWithRestrictions"
    },
    {
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "RunInstances"
        }
      },
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:volume/*",
```

```
            "arn:aws:ec2:*:*:instance/*"
        ],
        "Effect": "Allow",
        "Sid": "AllowCreateTagsOnlyLaunching"
      }
    ]
}
```

This policy statement allows users in this group to launch an EC2 instance, so long as the instance is a **t3.medium** and the instance has been populated with the following tags:

Key	Value
Application	Accounts Payable
Department	Finance
Environment	Development

40. Choose the **Cancel** link at the bottom of the pop-up window.

Task 3.2: Add the awsstudent user to the APDevelopers Group

Now that you have reviewed the user group and inline policy, you will add the user awsstudent to the group. This will allow the user to create EC2 instances, so long as they are compliant with the policy conditions.

41. Choose **Users** tab.

42. Select on **awsstudent** User Name.

43. Choose **Groups** tab.

44. Select on **Add user to groups**

45. Check ☑ **APDevelopers**

46. Choose **Add to Groups**

Task 3.3: Verify Preventative Controls

With awsstudent now belonging to the APDevelopers group, you will want to confirm that the inline policy is restricting permissions as intended.

47. Open the **EC2 Console** by selecting **Services** and typing `ec2` in the filter box.

48. Choose **EC2**.

49. In the left menu, select **Instances**.

50. Select **Launch Instance**

51. Next to **Amazon Linux 2 AMI**, select **Select**

52. Select an Instance Type of **t3.medium**.

53. Choose **Next: Configure Instance Details**

    On the Instance Details page, you will want to identify the VPC to deploy your new instance in.

54. In the **Network** field, select **Lab-VPC**.

55. Choose **Next: Add Storage**

    No adjustments are required for the storage options for the instance.

56. Choose **Next: Add Tags**

    You will want to add tags to these instances to make sure they are compliant

with your organizations published architecture standards. This will avoid the instance from flagging as non-compliant based on the Config rules you have implemented.

57. Choose **Add Tag**
58. Enter a Key of `Name` and a Value of `My Server`
59. Choose **Add another tag**
60. Enter a Key of `Environment` and a Value of `Dev`
61. Choose **Add another tag**
62. Enter a Key of `Application` and a Value of `Accounts Payable`
63. Choose **Add another tag**
64. Enter a Key of `Department` and a Value of `Finance`
65. Choose **Next: Configure Security Group**

    You will use a security group that is already in place in your environment for this Dev instance.

66. Select the **Select an existing security group** radio button.
67. Select the checkbox for the security group with a description of ☑ **Security Group for AP Test App Server**
68. Choose **Review and Launch**

    You will likely receive a warning that you will not be able to connect to this instance. That is ok, as you would not be connecting to this instance using SSH.

69. Choose **Continue**.

70. Verify your configuration choices on the Review page and select **Launch**

    A pop-up window will appear asking you to select a key pair.

71. In the first dropdown of the popup window, select **Proceed without a key pair**.

72. Check ☑ to acknowledge that you will not be able to connect to the instance.

73. Choose **Launch Instances**

    You will receive an error message that launch failed. Do you know why?

    The reason for failure is that you entered all of the necessary tags, but entered an incorrect value for the Environment tag. You need to enter **Development** instead of Dev.

74. Choose **Back to Review Screen** so that you can modify the configuration.

75. Choose the **Edit tags** hyperlink towards the bottom of the page.

    Now you can modify the incorrect tag that was defined earlier.

76. Change the Environment tag value from Dev to **Development**.

77. Choose **Review and Launch**

78. Choose **Launch**

    The pop-up window asking you to select a key pair will appear again.

79. In the first dropdown of the popup window, select **Proceed without a key pair**.

80. Check ☑ to acknowledge that you will not be able to connect to the instance.

81. Choose **Launch Instances**

Because you have specified the correct required tag values, you should now receive a message that reads **Your instances are now launching**.

82. Choose **View Instances**

    Here you can view all of the instances running in your account, including a new instance with the appropriate tags that are compliant with your AWS Config rules established earlier in the lab.

Task 4: Review and Remediate AWS Config Findings

Now that you have created your AWS Config rules and allowed some time for Config to evaluate the resources deployed in your account, you will review the results of each rule to determine which resources are non-compliant.

Task 4.1: Review Compliance with RequiredTagsCompliance Rule

83. Open the **AWS Config Console** by selecting **Services** and typing `config` in the filter box.

84. Choose **Config**.

85. Choose **Rules** in the top section of the left menu. **NOTE:** Be careful not to select Rules under Aggregated View.

86. On the Rules dashboard, choose the **required-tags** link.

    The instances that were not tagged in accordance to the rules defined are listed at the bottom of the page.

87. Review the details for the first instance by selecting the hyperlink in the **Resource ID** column.

88. Expand the **Tags** section to see all of the tags applied to the instance.

89. Choose **Manage resource**

90. Choose **Actions** and select **View details**.

91. Choose the **Tags** tab.

92. Choose **Manage Tags**

93. Using the table below as a guide, correct any missing or incorrect tags on the instance.

Name	Department	Application	Environment
App-Server-Prod	`Finance`	`Accounts Payable`	`Production`
DB-Server-Prod	`Finance`	`Accounts Payable`	`Production`
App-Server-Test	`Finance`	`Accounts Payable`	`Test`
DB-Server-Test	`Finance`	`Accounts Payable`	`Test`

    NOTE: Use **Add Tag** to add missing Tags if needed.

94. Choose **Save** once you have updated or added the required tags.

95. Choose the **gear icon** (upper right).

96. In the **Tag Columns**, select the drop down and choose check boxes beside

Application , Environment, and Department

97. Choose **Confirm**

    Displaying the tag columns will make it easier to review the tags on all of your instances for any missing or incorrect tags.

98. In the search bar above the listed instance, remove the current search tag to list all of the instances.

    You should see 5 instances deployed in this region.

99. Using the displayed tags, correct or populate all required tags on your instances based on the previous table.

    - Choose each missing or incorrect tag value.
        - Enter the appropriate value from the previous table.
        - Select the checkmark for each tag once populated correctly.

100. Select the two instances tagged as Test.

101. Choose **Instance state**

102. Select **Reboot instance**.

103. Choose **Reboot**

     The action of rebooting the instances that you have modified the tags on will, force Config to re-evaluate these instances for compliance.

     NOTE: You might have to reboot all instances or re-evaluate rules a couple of times.

Task 4.2: Review Compliance with

ProductionInstanceType Rule

104. Open the **AWS Config Console** by selecting **Services** and typing `config` in the filter box.

105. Choose **Config**.

106. Choose **Rules** in the left menu.

107. On the Rules dashboard, select the **ProductionInstanceType** link.

     The instances that were deployed with a non-compliant instance type are listed at the bottom of the page.

108. Review the details for the first instance by selecting the hyperlink in the **Resource ID** column.

109. Choose **Manage resource**

110. In the search bar above the listed instance, remove the current search tag to list all of the instances.

     You should see that your Production EC2 instances are now in a stopped or stopping state, as your Config rule was configured for auto remediation. You can now resize the necessary instances to get them back into a running state, but it is best to focus on the Production instances first.

111. Using the **Environment** tag, select one of the **Production** instances.

112. Choose **Instance state**

113. Select **Stop instance**

114. Confirm by choosing on **Stop**

115. Choose **Actions**

116. Choose **Instance Settings** in the dropdown.

117. Select **Change Instance Type**.

118. Set the Instance Type to **t3.large**.

119. Choose **Apply**

    Now that you have selected a compliant Instance Type, you will need to restart the instance.

120. Choose **Instance state**

121. Select **Start instance**

    The Instance State should change to pending and will move to running shortly.

122. Repeat the previous steps to resize and restart your other Production instance.

    Your production instances should now be resized and compliant with both Config rules that you have configured.

Task 4.3: Final Compliance Check

123. Open the **AWS Config Console** by selecting **Services** and typing `config` in the filter box.

124. Choose **Config**.

125. Select on **View dashboard**

126. Choose **Rules** in the left menu.

    You should now see that all resources are compliant with both of the Config

rules that were created.

127. On the Rules dashboard, the **Compliance status** should have changed to **Compliant**.

Lab Complete

Congratulations! You have completed this lab. To clean up your lab environment, do the following:

128. To sign out of the AWS Management Console, choose **awsstudent** at the top of the console, and then select **Sign Out**.

129. On the Qwiklabs page, select **End**.

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

130. Return to the AWS Management Console.

131. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.

132. Choose **End Lab**

133. Choose **OK**

134. (Optional):

- Select the applicable number of stars ☆
- Type a comment
- Choose **Submit**

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

You may close the window if you don't want to provide feedback.