

Lab 5.1

AWS ●

EN-US ▾

► Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

Lab 5.1: Encrypting Data at Rest by Using AWS KMS

Lab overview and objectives

In this lab, you will use the AWS Key Management Service (AWS KMS) to encrypt data at rest. You will create an AWS KMS key, use it to encrypt objects stored in Amazon Simple Storage Service (Amazon S3), and use it to encrypt Amazon Elastic Block Store (Amazon EBS) volumes. You will also see how AWS CloudTrail provides an audit log of AWS KMS key usage and how disabling the key affects data access.

After completing this lab, you should be able to do the following:

- Create an AWS KMS customer managed key to encrypt and decrypt data at rest.
- Store an encrypted object in an S3 bucket by using an encryption key.
- Attempt public access and signed access to an encrypted S3 object.
- Monitor encryption key usage by using the CloudTrail event history.
- Encrypt the root volume of an existing Amazon Elastic Compute Cloud (Amazon EC2) instance.
- Disable and re-enable an AWS KMS key and observe the effects on data access.

Duration

This lab will require approximately **75 minutes** to complete.

AWS ●

EN-US ▾

► Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

Scenario

The following diagram shows the architecture that was created for you in AWS at the *beginning* of the lab. AWS Cloud resources include an empty S3 bucket named ImageBucket, and an EC2 instance named LabInstance, which has an unencrypted root EBS volume attached.

```
graph LR; User((User)) --> Internet((Internet)); Internet --- AWSCloud[AWS Cloud]; AWSCloud --- S3[Amazon S3]; S3 --- ImageBucket[ImageBucket]; AWSCloud --- AWSKMS[AWS KMS]; AWSCloud --- LabInstance[LabInstance]; LabInstance --- EBS[Amazon EBS]; EBS --- UnencryptedVolume[Unencrypted root EBS volume];
```

By the *end* of this lab, you will have created the architecture shown in the following diagram. This diagram also shows some of the actions you will take and communication that will occur between the Amazon S3,

EN-US ▾

By the *end* of this lab, you will have created the architecture shown in the following diagram. This diagram also shows some of the actions you will take and communication that will occur between the Amazon S3, Amazon EBS, AWS KMS, and CloudTrail services during the lab.

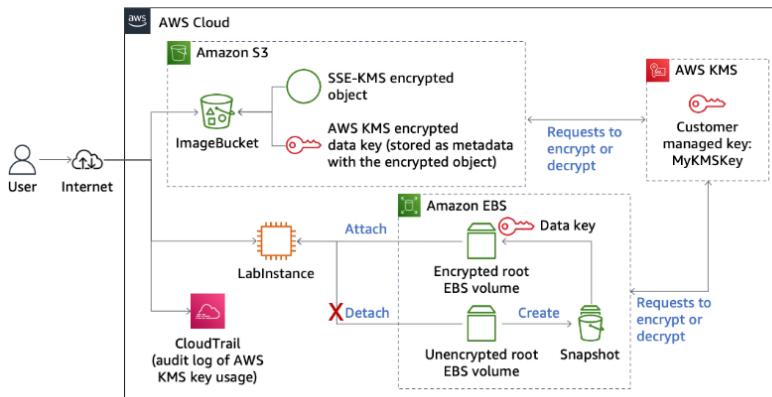


Diagram description: User connected to AWS Cloud resources through the internet. ImageBucket contains an object encrypted with server-side encryption with AWS KMS, known as SSE-KMS. Bucket also contains an encrypted data key, stored as metadata with the encrypted object. The unencrypted root EBS volume has been detached from LabInstance. A snapshot was created of this volume, and the snapshot was used to create an encrypted root EBS volume. The encrypted volume is attached to LabInstance. Requests to

EN-US ▾

Diagram description: User connected to AWS Cloud resources through the internet. ImageBucket contains an object encrypted with server-side encryption with AWS KMS, known as SSE-KMS. Bucket also contains an encrypted data key, stored as metadata with the encrypted object. The unencrypted root EBS volume has been detached from LabInstance. A snapshot was created of this volume, and the snapshot was used to create an encrypted root EBS volume. The encrypted volume is attached to LabInstance. Requests to encrypt or decrypt S3 objects and EBS volumes are sent to AWS KMS, where the customer managed key named MyKMSKey is stored. CloudTrail logs AWS KMS key usage as events. **End description.**

Accessing the AWS Management Console

1. At the top of these instructions, choose ▶ **Start Lab**.
 - The lab session starts.
 - A timer displays at the top of the page and shows the time remaining in the session.
 - 💡 **Tip:** To refresh the session length at any time, choose ▶ **Start Lab** again before the timer reaches 0:00.
 - Before you continue, wait until the circle icon to the right of the [AWS](#) link in the upper-left corner turns green. When the lab environment is ready, the AWS Details panel will also display.

2. To connect to the AWS Management Console, choose the [AWS](#) link in the upper-left corner, above the terminal window.
 - A new browser tab opens and connects you to the console.
 - 💡 **Tip:** If a new browser tab does not open, a banner or icon is usually at the top of your browser.

EN-US

Task 1: Creating an AWS KMS key

In this task, you will create a customer managed AWS KMS key. The AWS KMS key that you create will be used later in the lab to generate, encrypt, and decrypt *data keys*. The data keys will be shared with other AWS services, including Amazon S3 and Amazon EC2. The data keys will be used to encrypt the actual data stored in an S3 bucket and on EBS volumes.

3. Create a customer managed AWS KMS key.

- o In the AWS Management Console, in the search box to the right of **Services**, search for **KMS** and choose **Key Management Service** to open the AWS KMS console.
- o In the navigation pane, choose **Customer managed keys**.
- o **Tip:** To open the navigation pane, choose the **≡** menu icon in the upper-left corner.
- o Choose **Create key**.
- o For **Key type**, choose **Symmetric**.
- o **Note:** Symmetric keys never leave AWS KMS unencrypted. The key that you create will be a 256-bit secret key.
- o Choose **Next**.
- o For **Alias**, enter `MyKMSKey`
- o Choose **Next**.
- o For **Key administrators**, search for and choose the `voclabs` role.

Note: Key administrators are users or roles that will *manage access* to the encryption key. You are

EN-US

bit secret key.

- o Choose **Next**.
- o For **Alias**, enter `MyKMSKey`
- o Choose **Next**.
- o For **Key administrators**, search for and choose the `voclabs` role.
- o **Note:** Key administrators are users or roles that will *manage access* to the encryption key. You are currently logged in to the console with the `voclabs` role. This user role is displayed in the upper-right corner of the console, to the left of the AWS Region.
- o Choose **Next**.
- o On the **Define key usage permissions** page, search for and choose the `voclabs` role again.
- o **Note:** Key users are the users or roles that can *use the key* to encrypt and decrypt data.
- o Choose **Next**.
- o At the bottom of the **Review** page, choose **Finish**.

Your key displays in the list of customer managed keys.

Note: If you attempted this lab within the past 7 days, you might see an older key or older keys in this list, with the status *Pending deletion*. The minimum amount of time that must transpire between when you request that a key be scheduled to be deleted and when it's actually deleted is 7 days.

In this task, you created an AWS KMS key, which you will use in the next task.

AWS ●

EN-US ▾

Task 2: Storing an encrypted object in an S3 bucket

In this task, you will upload an image file to an S3 bucket. You will encrypt the file by using the encryption key that you created in the previous task.

4. To download the file named **clock.png** to your computer, choose the following link: [Download](#).

5. Locate the S3 bucket and analyze its encryption settings.

- o In the search box to the right of **Services**, search for and choose **S3** to open the Amazon S3 console.
- o In the navigation pane, choose **Buckets**.
- o Choose the link for the bucket name that contains **imagebucket**.
- o Choose the **Properties** tab.
- o In the **Default encryption** section, notice that the setting is currently enabled.

When this setting is enabled, new objects that are stored in the bucket are automatically encrypted.
This setting is enabled by default.

6. Upload a file to the bucket and store it as an encrypted object.

- o At the top of the page, choose the **Objects** tab.

AWS ●

EN-US ▾

6. Upload a file to the bucket and store it as an encrypted object.

- o At the top of the page, choose the **Objects** tab.
- o Choose **Upload**.
- o Choose **Add files**.
- o Browse to and select the **clock.png** file on your computer.
- o Expand the **Properties** section, and configure the following:
 - In the **Server-side encryption settings** section, choose **Specify an encryption key**.
 - **Encryption key type:** Choose **AWS Key Management Service key (SSE-KMS)**.
 - **AWS KMS key:** Select **Choose from your AWS KMS keys**.
 - **AWS KMS key** dropdown menu: Choose **MyKMSKey**.
- o At the bottom of the page, choose **Upload**.
- o After the upload completes, choose **Close** in the upper-right corner.

Notice that **clock.png** is now listed as an object in the bucket.

The following diagram and description explain what transpired when you uploaded the file to Amazon S3.

```

graph LR
    User((User)) -- 1 --> ImageBucket[ImageBucket]
    ImageBucket -- 2 --> KMS[AWS KMS]
    subgraph KMS [AWS KMS]
        direction TB
        KeyIcon((Key icon))
        KeyText[Customer managed key: MyKMSKey]
    end
    KMS -- 3 --> User
    ImageBucket -- 5 --> KMS

```

The diagram illustrates the process of uploading a file to an S3 bucket and encrypting it using an AWS KMS key. It shows a flow from a User to an ImageBucket, then to AWS KMS, and finally back to the User. Step 1 shows the User interacting with the ImageBucket. Step 2 shows the data being sent to AWS KMS. Step 3 shows the User receiving the encrypted data. Step 5 shows the ImageBucket sending the data to AWS KMS for encryption.

AWS ●

EN-US ▾

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

Step	Explanation
1	The user requested to upload a file and store it as an encrypted object in ImageBucket.
2	Amazon S3 requested a data key from AWS KMS to use to encrypt the file.
3	AWS KMS generated a plaintext data key and encrypted the data key by using the customer managed key MyKMSkey.
4	AWS KMS sent both copies of the data key to Amazon S3.
5	Amazon S3 encrypted the object by using the plaintext data key, stored the object, and then deleted the plaintext data key. The encrypted key was kept in the object metadata.

7. Analyze the encryption settings on the object.

- Choose the link for the **clock.png** object.

AWS ●

EN-US ▾

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

7. Analyze the encryption settings on the object.

- Choose the link for the **clock.png** object.
- On the **Properties** tab, find the **Server-side encryption settings** section.

Notice that server-side encryption using SSE-KMS is enabled on this object.

In this task, you analyzed the encryption settings on the bucket. You then uploaded an object to the bucket. Although the bucket's settings do not automatically encrypt objects that you upload, you had the *option* to encrypt an object that you uploaded.

Task 3: Attempting public access to the encrypted object

In this task, you will attempt to access the encrypted object by using the S3 object URL. This URL is a method to access an S3 object from the public internet.

8. Attempt to open the image by using the S3 object URL.

- In the **Object overview** section at the top of the page, copy the **Object URL** to your clipboard.
- Note:** Notice that the URL is in the following format: <https://s3.amazonaws.com/>.
- Paste the object URL into a new browser tab, and attempt to load the page.

You receive the following *Access Denied* error.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

EN-US

8. Attempt to open the image by using the S3 object URL.

- o In the **Object overview** section at the top of the page, copy the **Object URL** to your clipboard.
Note: Notice that the URL is in the following format: <https://s3.amazonaws.com/>.
- o Paste the object URL into a new browser tab, and attempt to load the page.

You receive the following *Access Denied* error.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>TWRYYNND3JNNQFCY</RequestId>
<HostId>IWj3DNg9bmYdzj+UPANJ9xKBpk/sKAv044+rcW5J5nZG36pNtWn49n6BoiYIoNa19v3vkXLO/k=</HostId>
</Error>
```

Note: The S3 object URL is one method to provide public internet access to an S3 object. However, by default, when you create an S3 bucket, public access to objects is not allowed. The default *Block all public access* security setting is applied to this bucket and all objects stored in it.

- o Keep the browser tab open, because you will return to this page in a moment. Return to the browser tab where the Amazon S3 console is open.

9. Modify the public access permissions for the *bucket*.

- o In the breadcrumbs, which are located in the upper-left corner of the page, choose the bucket name, which contains **imagebucket**.
Choose the **Permissions** tab

EN-US

8. Modify the public access permissions for the *bucket*.

- o In the breadcrumbs, which are located in the upper-left corner of the page, choose the bucket name, which contains **imagebucket**.
- o Choose the **Permissions** tab.
- o In the **Block public access** section, choose **Edit**.
- o Clear **Block all public access**.
- o Choose **Save changes**.
- o When prompted, enter **confirm** and choose **Confirm**.

10. Modify the access settings for the *object*.

- o Choose the link for the bucket name that contains **imagebucket** and choose the **Permissions** tab.
- o Under **Object Ownership** choose edit. Choose **ACLs enabled**.
- o Select the checkbox for the acknowledgement. Keep **Bucket owner preferred** selected, and choose **Save changes**.
- o Choose the **Objects** tab, and then select **clock.png**.
- o Choose **Actions > Make public using ACL**.
- o Choose **Make public**.
A banner displays to indicate that the public access settings were successfully edited.
- o Choose **Close**.

AWS ●

EN-US

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

11. Test access to the object again by using the S3 object URL.

- o Return to the browser tab that you opened earlier and attempted to load the S3 object URL.
- o Refresh the page.

You now see a different error display. Instead of the Access Denied error that you saw earlier, you now see the following *Invalid Argument* error.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
  <Code>InvalidArgument</Code>
  <Message>Requests specifying Server Side Encryption with AWS KMS managed keys require AWS Signature Version 4.</Message>
  <ArgumentName>Authorization</ArgumentName>
  <ArgumentValue>null</ArgumentValue>
  <RequestId>R2QP3N7E4P7VZOB</RequestId>
  <HostId>bJgB1+STQ/#fs#6eINOAq1qcdJhwjhTe4/eJ038tfSP2ifdYYAk0650K1DV2tNPP620Sv?Bm+H=</HostId>
</Error>
```

Analysis: Access to the object is now granted to the public internet (through the object URL); however, because the image is encrypted, you are still not able to view it.

This is a good thing, because it demonstrates an important reason that you might choose to encrypt data. In cases where access to data might be accidentally granted to someone who should not have access, the person won't be able to read the data unless they *also* have access to the encryption key. The encryption key is needed to decrypt the data and make it human readable again.

The error message indicates that *Requests specifying Server Side Encryption with AWS KMS managed keys require AWS Signature Version 4*. Signature Version 4 is the process to add authentication information to AWS requests.

AWS ●

EN-US

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

- Close the browser tab where you loaded the S3 object URL.

In this task, you experienced how the encrypted object that is stored in Amazon S3 was first not accessible through the internet. Then, you discovered that, even when you made the object available to the internet, you weren't able to successfully access the object. This happened because requests that specify SSE-KMS require AWS Signature Version 4, and you did not sign the request.

Task 4: Attempting signed access to the encrypted object

In this task, you attempt to decrypt the encrypted object from within the Amazon S3 console as an authenticated user.

12. Open the stored encrypted object.

- o In the Amazon S3 console, make sure you are on the **Objects** tab for the bucket with **imagebucket** in its name.
- o Select **clock.png**, and then choose **Open**.

The following image opens in a new tab or window.



EN-US ▾

- In the Amazon S3 console, make sure you are on the **Objects** tab for the bucket with **imagebucket** in its name.
- Select **clock.png**, and then choose **Open**.

The following image opens in a new tab or window.



13. Analyze the URL that was used to successfully decrypt the object.

EN-US ▾

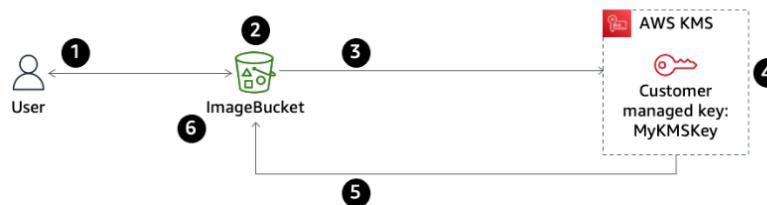
13. Analyze the URL that was used to successfully decrypt the object.

- Notice the URL in the browser tab where the image is open. The format is similar to the following (where the **...** values contain credentials information).

```
https://c46588a73057211728517t1w501832962033-imagebucket-gmnrbh84muj.s3.us-east-1.amazonaws.com/clock.png?response-content-disposition=inline&X-Amz-...&X-Amz-...&X-Amz-...&X-Amz-...&X-Amz-...&X-Amz-...&X-Amz-...
```

- Close the browser tab where the image is loaded, and return to the Amazon S3 console.

The following diagram and description explain what transpired when you requested to open the stored encrypted object.



Step	Explanation
1	You requested to open the object.

AWS

EN-US

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

Step	Explanation
1	You requested to open the object.
2	Next, Amazon S3 noticed that the requested object was encrypted. Because you were authenticated to the AWS account when using the Amazon S3 console, the Signature Version 4 authentication information was automatically included in the request.
3	Amazon S3 then sent the encrypted copy of the <i>data key</i> that the object was encrypted with to AWS KMS.
4	AWS KMS then <i>decrypted the data key</i> by using the MyKMSKey AWS KMS key (which never leaves the AWS KMS service).
5	AWS KMS then sent the <i>plaintext data key</i> back to Amazon S3.
6	Finally, Amazon S3 decrypted the ciphertext of the data object, allowed you to open the object, and deleted the plaintext copy of the data key.

Task 5: Monitoring AWS KMS activity by using CloudTrail

In this task, you will access the CloudTrail event history to find events that are related to your encryption operations. The CloudTrail audit log functionality provides an important security feature, and it's a good idea to monitor how AWS KMS keys are used in your account.

AWS

EN-US

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

14. Access the CloudTrail event history.
 - o In the search box to the right of **Services**, search for and choose **CloudTrail** to open the CloudTrail console.
 - o In the navigation pane, choose **Event history**.

Tip: To open the navigation pane, choose the menu icon in the upper-left corner.

Note: CloudTrail provides an audit log of API calls that are made in the AWS account. The event history provides access to events from the last 90 days of account activity.

 - o Notice the column named **Event source**.

Each time that an API call to an AWS service occurs within the Region that you have selected, if that service reports such events to CloudTrail, then the event and the AWS service that reported the event are listed.
15. Filter the event history to display only events that the AWS KMS service reported.
 - o Choose the dropdown menu on the left that currently displays **Read-only**, and choose **Event source**.
 - o In the *Enter an event source* search box, search for `kms` and choose `kms.amazonaws.com`.
16. Analyze the **GenerateDataKey** event.
 - o Choose the link for the **GenerateDataKey** event name.
 - In the **Event details** section, observe the details of the event

AWS ●

EN-US

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

16. Analyze the GenerateDataKey event.

- o Choose the link for the **GenerateDataKey** event name.
- o In the **Event record** section, observe the details of the event.

The details are similar to the following image. In this image, essential details of the event record are highlighted.

Event record

```
{
  "eventVersion": "1.0",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAZZ74B66CPC34KTVI@amazon.com",
    "arn": "arn:aws:sts::674301556612:assumed-role/voclabs@amazon.com",
    "accountId": "674301556612",
    "accessKeyId": "ASIAJHUEVYPM7NQHVK2Q",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAZZ74B66CPC34KTVI",
        "arn": "arn:aws:iam::674301556612:role/voclabs",
        "accountId": "674301556612",
        "userName": "voclabs"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-03-09T19:57:21Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2022-03-09T21:12:01Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal"
}
```

AWS ●

EN-US

▶ Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

```
,
  "eventTime": "2022-03-09T21:12:01Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-east-1:674301556612:key/fe4a3f51-0aaa-4e90-b11a-ca181b2cdaab",
    "encryptionContext": {
      "aws:s3:arn": "arn:aws:s3:::c46588a7305721737683t1w674301556612-imagebucket-o2cvwzkvqvs6/clock.png"
    },
    "keySpec": "AES_256"
  },
  "responseElements": null,
  "requestID": "76cf3f1e-6dd4-411f-917a-bb8102019cd9",
  "eventID": "00dfabfe-64c0-451f-ad63-d08316dc13ce",
  "readOnly": true,
  "resources": [
    {
      "accountId": "674301556612",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-east-1:674301556612:key/fe4a3f51-0aaa-4e90-b11a-ca181b2cdaab"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "674301556612",
  "eventCategory": "Management",
  "sessionCredentialFromConsole": "true"
}
```

Analysis: This event was generated when you requested to upload the clock.png file to Amazon S3. Your action prompted AWS KMS to generate a new data key. If you go to the AWS KMS console and look at the key ID of the AWS KMS key that you created in task 1, you will find that it matches the keyId in the event record. Notice also that the event specifies the ARN of the object that Amazon S3 planned to encrypt with the data key, after it received the data key from AWS KMS. The event details also list the identity (principalId) of the user or role who made the request.

EN-US ▾

17. Analyze the Decrypt event.

- In the breadcrumbs at the top of the page, choose **Event history**.

Note: The events should still be filtered by the `kms.amazonaws.com` event source.

- Choose the link for the **Decrypt** event name.

- As you did previously, observe the details in the **Event record** section.

Analysis: This event was generated when you successfully opened the `clock.png` file from the Amazon S3 console. Notice that the record again details who made the request, which AWS KMS key provided the unencrypted *data key* back to Amazon S3, and which S3 object was decrypted with the plaintext data key.

CloudTrail logs all AWS KMS API activity. By evaluating these log entries, you might be able to determine the past usage of a particular AWS KMS key. If you want to be able to analyze events that occurred more than 90 days ago, you can create a CloudTrail trail.

Task 6: Encrypting the root volume of an existing EC2 instance

In this task, you will encrypt the root volume of an existing EC2 instance.

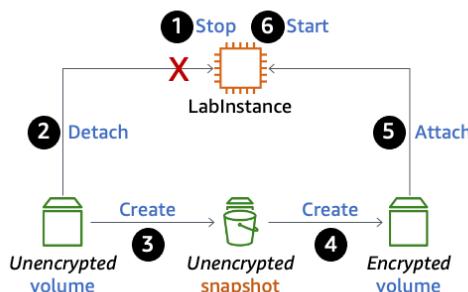
To encrypt an unencrypted volume, you need to complete multiple steps. The following diagram and description explain these steps.

EN-US ▾

Task 6: Encrypting the root volume of an existing EC2 instance

In this task, you will encrypt the root volume of an existing EC2 instance.

To encrypt an unencrypted volume, you need to complete multiple steps. The following diagram and description explain these steps.



Step	Explanation
1	Stop the instance.
2	Detach the volume.

EN-US ▾

Step	Explanation
1	Stop the instance.
2	Detach the volume.
3	Create a snapshot of the unencrypted root volume.
4	Create a new encrypted volume from that snapshot.
5	Swap the encrypted volume in as the new instance root volume.
6	And finally, start the instance.

18. Observe the current storage settings on an existing EC2 instance.

- o In the search box to the right of **Services**, search for and choose **EC2** to open the Amazon EC2 console.
 - o In the navigation pane, choose **Instances**.
 - o Choose the link for the **LabInstance** instance ID.
 - o Choose the **Storage** tab.
 - o In the **Block devices** section, notice that the volume that is attached indicates it is *not* encrypted. This is the root volume, which contains the guest OS installation.
- Note:** You can encrypt a volume attached to an EC2 instance when you first create the instance. You can also encrypt a volume that is attached to an existing EC2 instance, including to the root volume, but it requires a few more steps. You will complete those steps in this task.

EN-US ▾

19. Stop the instance.

- o In the breadcrumbs at the top of the page, choose **Instances**.
- o Select **LabInstance**, and choose **Instance state > Stop instance**.
- o To confirm the action, choose **Stop**.

20. Create a *snapshot* of the root EBS volume of the existing EC2 instance.

- o Choose the **Storage** tab.
- o In the **Block devices** section, choose the link for the **Volume ID**.
- o Choose the link for the **Volume ID** again.
- o Note the Availability Zone where the volume exists (for example, us-east-1a or us-east-1b).
- Important:** You will need this information in a moment.
- o Choose **Actions > Create snapshot**.
- o Choose **Add tag**, and add a tag with the following information:
 - **Key:** Enter `Name`
 - **Value:** Enter `Unencrypted Root Volume`
- o Choose **Create snapshot**.

21. Create an *encrypted* volume from the *unencrypted* snapshot.

AWS ●

EN-US ▾

► Start Lab ■ End Lab AWS Details Details X

Submit Submission Report Grades

21. Create an *encrypted* volume from the *unencrypted* snapshot.

- o In the navigation pane, under **Elastic Block Store**, choose **Snapshots**.
- o Choose the link for the **Unencrypted Root Volume** snapshot ID that you just created.
- o Wait until the **Snapshot status** shows *Completed*.
- o Notice that the encryption status of the snapshot is *Not encrypted*.
- o Choose **Actions > Create volume from snapshot**, and configure the following:
 - **Availability Zone:** Choose the Availability Zone where the existing volume exists.
 - Select **Encrypt this volume**.
 - **KMS key:** Choose **MyKMSKey**.
- o Choose **Create volume**.

22. Label the volumes.

- o In the navigation pane, under **Elastic Block Store**, choose **Volumes**.
- o Notice that two volumes are now listed.
- o For the volume with a **Volume state** of *In-use*, change the volume name:
 - Hover on the **Name** field, and choose the  pencil and paper icon.
 - In the **Edit Name** box, enter `old_unencrypted_root_volume`
 - Choose **Save**.
- o Follow the same steps to change the name of the volume with a **Volume state** of *Available* to `New encrypted root volume`

23. Swap the root volume that the EC2 instance uses.

- o Select **Old unencrypted root volume**, and then choose **Actions > Detach volume**.
- o To confirm, choose **Detach**.
- o Select **New encrypted root volume**, and then choose **Actions > Attach volume** and configure the following:
 - **Instance:** Choose **(LabInstance) (stopped)**.
 - **Device name:** Enter `/dev/xvda`
 - **Note:** This is the device name where the existing instance expects to find the root volume.
- o Choose **Attach volume**.

24. Notice that the root volume is now encrypted.

- o Return to the **Instances** screen, and select **LabInstance**.
- o Choose the **Storage** tab, and notice that the attached volume is now encrypted and has a AWS KMS key ID.
- **Note:** You might need to refresh the page to see the latest information for the attached volume.

EN-US ▾

In this task, you encrypted the root volume of an existing instance so that the data on it is now more secure.

Task 7: Disabling the encryption key and observing the effects

In this task, you will temporarily disable the AWS KMS key that you previously used to encrypt an object in Amazon S3 and encrypt an EBS volume. You will then observe the effects that disabling the key has on accessing encrypted data.

25. Disable the AWS KMS key that you used to encrypt the S3 object and EBS volume.

- o Navigate to the AWS KMS console.
- o In the navigation pane, choose **Customer managed keys**.
- o Select **MyKMSKey**, and then choose **Key actions > Disable**.
- o To confirm the action, select *Confirm that you want to disable this key*, and then choose **Disable key**.

26. Try to start the EC2 instance.

- o Navigate to the Amazon EC2 console.

EN-US ▾

- o Select **LabInstance**, and then choose **Instance state > Start instance**.
- o Use the refresh icon  to monitor the **Instance state**.

It will quickly fail to start and the instance state will change from *Pending* to *Stopped* again.

27. Try to access the S3 object by using the object URL.

- o Navigate to the Amazon S3 console.
- o In the navigation pane, choose **Buckets**.
- o Choose the link for the bucket name that contains **imagebucket**.
- o Select **clock.png**, and then choose **Open**.

The image fails to open, and a **KMS.DisabledException** error displays, even though this approach to decrypting the object was successful for you earlier in the lab.

28. Analyze the CloudTrail event history.

- o Navigate to the CloudTrail console.
- o In the navigation pane, choose **Event history**.
- o Choose the link for the **DisableKey** event name.
- o Examine the event details.

The details show that you disabled the AWS KMS key a few minutes ago.

- o Examine the **StartInstances** event, which occurred just after the **DisableKey** event.

The request to start the instance was successful. However, as you experienced, the instance never

EN-US ▾

The request to start the instance was successful. However, as you experienced, the instance never reached the *Running* state.

- Examine the **CreateGrant** event, which occurred just after the **StartInstances** event. The event record is similar to the following

```
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-east-1",
"sourceIPAddress": "AWS Internal",
"userAgent": "AWS Internal",
"errorCode": "DisabledException",
"errorMessage": "arn:aws:kms:us-east-1:259696676954:key/0599d416-76db-4055-8c46-7d3f1572a632 is disabled.",
"requestParameters": {
    "retiringPrincipal": "ec2.us-east-1.amazonaws.com",
    "operations": [
        "Decrypt"
    ],
    "keyId": "arn:aws:kms:us-east-1:259696676954:key/0599d416-76db-4055-8c46-7d3f1572a632",
    "granteePrincipal": "arn:aws:sts::259696676954:assumed-role/aws:ec2-infrastructure/i-0029c7208e05f2654",
    "constraints": {
        "encryptionContextSubset": {
            "aws:eb:sid": "vol-018e3301bf4bb5132"
        }
    }
}
```

Notice that the event details reveal an error. The EC2 instance start process noticed that the root volume was encrypted with the MyKMSKey, so Amazon EC2 contacted AWS KMS to provide the plaintext data key so it could decrypt the volume. But AWS KMS denied the request because the AWS KMS key that was used to encrypt the data key that the EBS volume was encrypted with was disabled. The instance can't start because the files on the root volume that the guestOS needs to start couldn't be decrypted.

29. Return to the AWS KMS console, and re-enable the MyKMSKey customer managed AWS KMS key.

EN-US ▾

AWS KMS key that was used to encrypt the data key that the EBS volume was encrypted with was disabled. The instance can't start because the files on the root volume that the guestOS needs to start couldn't be decrypted.

29. Return to the AWS KMS console, and re-enable the MyKMSKey customer managed AWS KMS key.

30. Return to the Amazon EC2 console, and start the LabInstance again.

The instance successfully starts now.

Important: Before you submit your work for the lab, wait for the **Instance state** to display *Running*.

Submitting your work

31. To record your progress, choose **Submit** at the top of these instructions.

32. When prompted, choose **Yes**.

After a couple of minutes, the grades panel appears and shows you how many points you earned for each task. If the results don't display after a couple of minutes, choose **Grades** at the top of these instructions.