Log in to the AWS console using the cloud\_user credentials provided. Once inside the AWS account, make sure you are using us-east-1 (N. Virginia) as the selected region.

**Hint:** When copying and pasting code into Vim from the lab guide, first enter *:set paste* (and then *i* to enter insert mode) to avoid adding unnecessary spaces and hashes.

### Create a Trust Policy and Role Using the AWS CLI

#### Obtain the labreferences.txt File

1. Navigate to S3.
2. From the list of buckets, open the one that contains the text s3bucketlookupfiles in the middle of its name.
3. Select the labreferences.txt file.
4. Click **Actions** > **Download**.
5. Open the labreferences.txt file, as we will need to reference it throughout the lab.

#### Log in to Bastion Host and Set the AWS CLI Region and Output Type

1. Navigate to **EC2** > **Instances**.
2. Copy the public IP of the Bastion Host instance.
3. Open a terminal, and log in to the bastion host via SSH:

ssh cloud\_user@<BASTION\_HOST\_PUBLIC\_IP>

For more information on how to connect to a Linux instance using SSH, please refer to the [AWS Documentation](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html). For more information on how to connect to a Linux instance using Putty, please refer to [Connect to your Linux instance from Windows using PuTTY](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html).

1. Enter the password provided for it on the lab page.
2. Run the following command:

[cloud\_user@bastion]$ aws configure

1. Press **Enter** twice to leave the AWS Access Key ID and AWS Secret Access Key blank.
2. Enter us-east-1 as the default region name.
3. Enter json as the default output format.

#### Create IAM Trust Policy for an EC2 Role

1. Create a file called trust\_policy\_ec2.json:

[cloud\_user@bastion]$ vim trust\_policy\_ec2.json

1. To avoid adding unnecessary spaces or hashes, type :set paste and then i to enter insert mode.
2. Paste in the following content:
3. {
4. "Version": "2012-10-17",
5. "Statement": [
6. {
7. "Effect": "Allow",
8. "Principal": {"Service": "ec2.amazonaws.com"},
9. "Action": "sts:AssumeRole"
10. }
11. ]

}

1. Save and quit the file by pressing **Escape** followed by :wq!.

#### Create the DEV\_ROLE IAM Role

1. Run the following AWS CLI command:

[cloud\_user@bastion]$ aws iam create-role --role-name DEV\_ROLE --assume-role-policy-document file://trust\_policy\_ec2.json

#### Create an IAM Policy Defining Read-Only Access Permissions to an S3 Bucket

1. Create a file called dev\_s3\_read\_access.json:

[cloud\_user@bastion]$ vim dev\_s3\_read\_access.json

1. To avoid adding unnecessary spaces or hashes, type :set paste and then i to enter insert mode.
2. Enter the following content, replacing <DEV\_S3\_BUCKET\_NAME> with the bucket name provided in the labreferences.txt file:
3. {
4. "Version": "2012-10-17",
5. "Statement": [
6. {
7. "Sid": "AllowUserToSeeBucketListInTheConsole",
8. "Action": ["s3:ListAllMyBuckets", "s3:GetBucketLocation"],
9. "Effect": "Allow",
10. "Resource": ["arn:aws:s3:::\*"]
11. },
12. {
13. "Effect": "Allow",
14. "Action": [
15. "s3:Get\*",
16. "s3:List\*"
17. ],
18. "Resource": [
19. "arn:aws:s3:::<DEV\_S3\_BUCKET\_NAME>/\*",
20. "arn:aws:s3:::<DEV\_S3\_BUCKET\_NAME>"
21. ]
22. }
23. ]

}

1. Save and quit the file by pressing **Escape** followed by :wq!.
2. Create the managed policy called DevS3ReadAccess:

[cloud\_user@bastion]$ aws iam create-policy --policy-name DevS3ReadAccess --policy-document file://dev\_s3\_read\_access.json

1. Copy the policy ARN in the output, and paste it into the labreferences.txt file — we'll need it in a minute.

### Create Instance Profile and Attach Role to an EC2 Instance

#### Attach Managed Policy to Role

1. Attach the managed policy to the role, replacing <DevS3ReadAccess\_POLICY\_ARN> with the ARN you just copied:

[cloud\_user@bastion]$ aws iam attach-role-policy --role-name DEV\_ROLE --policy-arn "<DevS3ReadAccess\_POLICY\_ARN>"

1. Verify the managed policy was attached:

[cloud\_user@bastion]$ aws iam list-attached-role-policies --role-name DEV\_ROLE

#### Create the Instance Profile and Add the DEV\_ROLE via the AWS CLI

1. Create instance profile named DEV\_PROFILE:

[cloud\_user@bastion]$ aws iam create-instance-profile --instance-profile-name DEV\_PROFILE

1. Add role to the DEV\_PROFILE called DEV\_ROLE:

[cloud\_user@bastion]$ aws iam add-role-to-instance-profile --instance-profile-name DEV\_PROFILE --role-name DEV\_ROLE

1. Verify the configuration:

[cloud\_user@bastion]$ aws iam get-instance-profile --instance-profile-name DEV\_PROFILE

#### Attach the DEV\_PROFILE Role to an Instance

1. In the AWS console, navigate to **EC2** > **Instances**.
2. Copy the instance ID of the instance named Web Server instance and paste it into the labreferences.txt file — we'll need it in a second.
3. In the terminal, attach the DEV\_PROFILE to an EC2 instance, replacing <LAB\_WEB\_SERVER\_INSTANCE\_ID> with the Web Server instance ID you just copied:

[cloud\_user@bastion]$ aws ec2 associate-iam-instance-profile --instance-id <LAB\_WEB\_SERVER\_INSTANCE\_ID> --iam-instance-profile Name="DEV\_PROFILE"

1. Verify the configuration (be sure to replace <LAB\_WEB\_SERVER\_INSTANCE\_ID> with the Web Server instance ID again):

[cloud\_user@bastion]$ aws ec2 describe-instances --instance-ids <LAB\_WEB\_SERVER\_INSTANCE\_ID>

This command's output should show this instance is using DEV\_PROFILE as an IamInstanceProfile. Verify this by locating the IamInstanceProfile section in the output, and look below to make sure the "Arn" ends in /DEV\_PROFILE.

### Test S3 Permissions via the AWS CLI

1. In the AWS console, copy the public IP of the Web Server instance.
2. Open a new terminal.
3. Log in to the web server instance via SSH:

ssh cloud\_user@<WEB\_SERVER\_PUBLIC\_IP>

1. Use the same password for the bastion host provided on the lab page.
2. Verify the instance is assuming the DEV\_ROLE role:

[cloud\_user@webserver]$ aws sts get-caller-identity

We should see DEV\_ROLE in the Arn.

1. List the buckets in the account:

[cloud\_user@webserver]$ aws s3 ls

Copy the entire name (starting with cfst) of the bucket with s3bucketdev in its name.

1. Attempt to view the files in the s3bucketdev- bucket, replacing <s3bucketdev-123> with the bucket name you just copied:

[cloud\_user@webserver]$ aws s3 ls s3://<s3bucketdev-123>

We should see a list of files.

### Create an IAM Policy and Role Using the AWS Management Console

#### Create Policy

1. In the AWS console, navigate to **IAM** > **Policies**.
2. Click **Create policy**.
3. Click the **JSON** tab.
4. Paste the following text as the policy, replacing <PROD\_S3\_BUCKET\_NAME> with the bucket name provided in the labreferences.txt file:
5. {
6. "Version": "2012-10-17",
7. "Statement": [
8. {
9. "Sid": "AllowUserToSeeBucketListInTheConsole",
10. "Action": ["s3:ListAllMyBuckets", "s3:GetBucketLocation"],
11. "Effect": "Allow",
12. "Resource": ["arn:aws:s3:::\*"]
13. },
14. {
15. "Effect": "Allow",
16. "Action": [
17. "s3:Get\*",
18. "s3:List\*"
19. ],
20. "Resource": [
21. "arn:aws:s3:::<PROD\_S3\_BUCKET\_NAME>/\*",
22. "arn:aws:s3:::<PROD\_S3\_BUCKET\_NAME>"
23. ]
24. }
25. ]

}

1. Click **Next: Tags**.
2. Click **Next: Review**.
3. Enter **ProdS3ReadAccess** as the policy name.
4. Click **Create policy**.

#### Create Role

1. Click **Roles** in the left-hand menu.
2. Click **Create role**.
3. Under Choose a use case, select **EC2**.
4. Click **Next: Permissions**.
5. In the Filter policies search box, enter **ProdS3ReadAccess**.
6. Click the checkbox to select **ProdS3ReadAccess**.
7. Click **Next: Tags**.
8. Click **Next: Review**.
9. Give it a Role name of **PROD\_ROLE**.
10. Click **Create role**.

### Attach IAM Role to an EC2 Instance Using the AWS Management Console

1. Navigate to **EC2** > **Instances**.
2. Select the Web Server instance.
3. Click **Actions** > **Security** > **Modify IAM role**.
4. In the IAM role dropdown, select **PROD\_ROLE**.
5. Click **Save**.

#### Test the Configuration

1. Open the existing terminal connected to the Web Server instance. (You may need to reconnect if you've been disconnected.)
2. Determine the identity currently being used:

[cloud\_user@webserver]$ aws sts get-caller-identity

This time, we should see PROD\_ROLE in the Arn.

1. List the buckets:

[cloud\_user@webserver]$ aws s3 ls

1. Copy the entire name (starting with cfst) of the bucket with s3bucketprod in its name.
2. Attempt to view the files in the s3bucketprod- bucket, replacing <s3bucketprod-123> with the bucket name you just copied:

[cloud\_user@webserver]$ aws s3 ls s3://<s3bucketprod-123>

It should list the files.

1. In the aws s3 ls command output, copy the entire name (starting with cfst) of the bucket with s3bucketsecret in its name.
2. Attempt to view the files in the <s3bucketsecret-123> bucket, replacing <s3bucketsecret-123> with the bucket name you just copied:

[cloud\_user@webserver]$ aws s3 ls s3://<s3bucketsecret-123>

This time, our access will be denied — which means our configuration is properly set up.