Task 1: Configuring an IAM group with policies and an IAM user

1. In the console, open the IAM service page.
2. Create an **IAM group** named AppDevelopers, and attach the following IAM policies to it:
    - **AmazonEC2ReadOnlyAccess**
    - **AWSCloud9EnvironmentMembe**
3. Create an **IAM user** and add the user to the *AppDevelopers* group.
    - **User name**: Nikhil
    - **Access type**: *AWS Management Console access*
    - **Custom password**: @ppD3veloper2020!
    - **Require password reset**: Clear this check box
    - Add Nikhil to the *AppDevelopers* group
    - In the **Success** screen, you can *optionally* choose **Download .csv** and save the file to your computer
    - Choose **Close**
4. While still logged in as the *voclabs* user, connect to the AWS Cloud9 IDE and set up the café web application.
    - Open the **AWS Cloud9** service page and under **DEVCafeServer**, choose **Open IDE**.
      The AWS Cloud9 IDE that run on an EC2 instance should now display.
    - In the Bash terminal window at the bottom of the screen, paste and run these three commands:

```
wget
https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod8-challenge/install-cafe-app.sh
chmod +x install-cafe-app.sh
./install-cafe-app.sh
```

5. Share the AWS Cloud9 environment with the *Nikhil* user.
    - In the top-right corner of the AWS Cloud9 IDE, choose **Share**.
    - In the **Share this environment** panel, under **Invite Members**, enterNikhil and choose **Invite**.
    - Choose **OK**, choose **OK** again, and then choose **Done**.

Task 2: Logging in as Nikhil and testing access

6. As *Nikhil*, log in to the AWS Management Console.

- In the browser tab where you are logged in as the *voclabs* user, open the **IAM** console, choose **Users**, and then choose **Nikhil**.
- Choose the **Security credentials** tab, and in the **Sign-in credentials** section, copy the **Console sign-in link**.
- Paste the link into an incognito or private browser tab (or other browser as explained in the previous tip).
- In the **Sign in as IAM user** screen, enter Nikhil's credentials and choose **Sign in**.
  - **IAM user name**: Nikhil
  - **Password**: @ppD3veloper2020!

7. Open the **Amazon EC2** console and in a browser tab, load the café web application.
   - Verify that you are in the correct **Region** (for example, *N. Virginia*) and switch to it, if necessary.
   - You should be able to view all the details of the EC2 instances.
   - Locate and copy the **IPv4 Public IP address** of the **aws-cloud9-DEVCafeServer** instance.
   - In a new browser tab, load http://<dev-public-ip-address>/cafe, where *<dev-public-ip-address>* is the IP address that you copied.
     The café website should display. Keep this browser tab open for later in the lab.
8. Test your Amazon EC2 access further by attempting to restart the web server.
   - Try to reboot the *aws-cloud9-DEVCafeServer* instance
   - To find the **Reboot instance** option, select the instance and look in the **Instance state** menu.

Accessing the Development server as Nikhil

9. Return to the browser tab where you are logged into the AWS Management Console as *Nikhil*.
10. Browse to the **AWS Cloud9** console, and connect to the AWS Cloud9 IDE on the *DEVCafeServer* EC2 instance.
    - From the **Services** menu, choose **AWS Cloud9**.
    - On the left, expand the menu by choosing the (menu icon), choose **Environments**. In the **Environments** drop-down it should say ***My environments***. Select the drop-down and choose ***Shared with me***. The **DEVCafeServer** environment is now listed.
    - Choose **Open**.
11. On the development instance of the café website, modify the main heading for the webpage.
    - Open the main webpage in the editor by going to the file browser, navigating to the DEVCafeServer/www/html/cafe directory, and double-clicking **index.php**.
    - Modify **line 13** So that it reads:
    - Café DEV Site

- To save the change, choose **File > Save** and in the browser, refresh the http://dev-public-ip-address/cafe/ webpage.
  Notice that, while acting as Nikhil, you changed the main heading of the webpage in the development environment.

12. Test the connectivity of the web application database.
    - In the café website, choose **Menu**.
    - Check what message displays.
13. As *Nikhil*, open the Systems Manager Parameter Store.
    - In the console, open the **Systems Manager** service.
    - From the menu on the left, choose **Application Management > Parameter Store**.
    - Check what message displays.
14. Nikhil alerts Sofía about the issue on the development server that's preventing him from improving the café web application. Sofía is concerned. She asks Nikhil to check if the *production* version of the website is experiencing the same issue.
15. As *Nikhil*, verify that the production café web application is working correctly.
    - Open the **Amazon EC2** console and copy the **IPv4 Public IP address** of the **PRODCafeServer** instance.
    - In a new browser window, load http://prod-public-ip-address/cafe/menu.php.
    - The webpage is now displaying properly and you are now able to place orders.

New business requirement: Configuring AWS account access for database administrators

---

Task 3: Configuring IAM for database administrator user access

In this task, you will work as *Sofía* to enable AWS access for Olivia.

16. Back in the browser where you are logged in as the *voclabs* user (Sofía), create an **IAM group** named DBAdministrators, with the following permissions:
    - **AmazonRDSReadOnlyAccess**
    - **AmazonSSMFullAccess**
17. In a real-world situation, Sofia would need to grant more than simply read only access to RDS to her database administrators. However, the permissions in this lab environment, do not allow you to attach the AmazonRDSFullAccess policy. Instead, for this lab, you should use the AmazonRDSReadOnlyAccess policy as a substitute.
18. Create an IAM user that's named Olivia with access to the **AWS Management Console**.
    - Set a custom password: Db@dministrat0r2020!
    - Clear the requirement to reset the password
19. Add Olivia to the **DBAdministrators** group.

Task 4: Logging in as the database administrator and resolving the database connectivity issue

In this task, you will work as *Olivia* to resolve the database issue that Nikhil identified. You will also work as *Sofia* to help Olivia resolve some issues.

20. As *Olivia*, log in to the AWS Management Console.
    - Choose **Nikhil @ <*account-number*>** in the top-right area of the console and choose **Sign Out**.
    - Then, choose **Log back in**.
      The **Sign in as IAM user** screen should display, with the *Account ID* pre-populated.
      Sign in with Olivia's credentials:
      - **IAM user name**: Olivia
      - **Password**: Db@dministrat0r2020!
21. Verify that the RDS database is running.
    - Open the **Amazon RDS** service page and choose **Databases**.
    - Verify that the **Status** of the database instance is *Available*.
22. Olivia observes that the database is running.
23. Open the **Amazon EC2** console and choose **Instances (running)**.Olivia tells Sofia that she can't access the EC2 instances, and Sofia goes back to the console to troubleshoot this issue.
24. You will now work as *Sofia* to review and update Olivia's access to AWS resources.
25. Return to the browser tab where you are logged in as the *voclabs* user (Sofia).
26. Open the **DBAdministrators** group, and attach these policies:
    - **AmazonEC2ReadOnlyAccess**
    - **IAMReadOnlyAccess**
27. Sofia realizes that Olivia needs some IAM permissions if she must access the details of the IAM role that's attached to the EC2 instance.
28. Still as the *voclabs* user, check which services and features Olivia used.
    - In the IAM console, open the **Olivia** user, and choose the **Access Advisor** tab
    - Notice that you can see which service areas that Olivia visited. Recent service activity usually appears within 4 hours (as stated in the Access Advisor details).
    - You might not see any **Last accessed data** for Olivia yet. This information enables you to more closely align access rights with the principle of least privilege.Sofia asks Olivia to check her Amazon EC2 access.
29. As *Olivia*, return to the browser tab where the Olivia user is logged in and refresh the instances page of the **Amazon EC2** console.
    - Olivia should now be able to access both running EC2 instances.
    - Select the **aws-cloud9-DEVCafeServer** instance.
    - In the **Details** tab, find **IAM role** and choose **CafeRole**.

- In the **Permissions** tab, expand the **AmazonSSMManagedInstanceCore** policy to see the permission details in **JSON**.
- Review the policy permissions.

30. Sofía asks Olivia to check whether the database user name is the source of the problem and to update it.
31. As *Olivia*, update the **dbUser** value in the Systems Manager Parameter Store.
32. In the web application on the development café server, refresh the **Menu** page.
    - If this webpage isn't already open, load http://<dev-public-ip-address>/cafe/menu.php in a browser (where *<dev-public-ip-address>* is the actual IPv4 public IP address of the **aws-cloud9-DEVCafeServer** instance).
    - The page should display correctly and successfully submit orders.

New business requirement: Refining IAM user access (Challenge #3)

Task 5: Using the IAM Policy Simulator and creating a custom IAM policy with the visual editor

33. Return to the browser window where you are logged in as the *voclabs* user (Sofía), and load this URL in a new browser tab: https://policysim.aws.amazon.com/
    The IAM Policy Simulator page should open.
34. Choose the **Olivia** user.
35. In the **IAM Policies** list, make sure that the **IAMReadOnlyAccess** policy is selected. However, *clear* the check boxes of the other policies.
36. In the **Policy Simulator** section, choose **Select service**. In the **Filter** search box, enter Ident and select **Identity and Access Management**.
37. Choose the **Select All** option (to the right of the **Select actions** menu), and then choose **Run Simulation**.
    - In the **Action Settings and Results** panel, a list of actions should display.
    - The **Permission** column displays Olivia's permissions for each action. The *IAMReadOnlyAccess* policy denies Olivia the permissions to perform **Add** or **Create** actions. However, scroll to find the actions that she *can* take.
    - The summary (at the top of the list) shows that Olivia is currently allowed to take *57 IAM actions*.
38. Return to the browser tab where you are logged in as the **voclabs** user (Sofía).
39. In the **IAM** console, choose **Policies** and then choose **Create Policy**.
40. In the **Visual editor** tab, configure the following settings.
    - Select **Choose a service**. Search for and choose **EC2**.
      - In the **Actions** search box, search for **IAM** and select **DescribeIamInstanceProfileAssociations**.
    - At the bottom of the screen, choose **Add additional permissions**.

- Select **Choose a service**. Search for and choose **IAM**.
  - In the **Actions** search box, search for **Get** and select the following actions
    - **GetPolicyVersion**
    - **GetRole**
    - **GetRolePolicy**
    - **GetInstanceProfile**
  - Back in the search box, search for **List** and select the following actions –
    - **ListAttachedRolePolicies**
    - **ListInstanceProfiles**
    - **ListInstanceProfilesForRole**
    - **ListPolicies**
    - **ListRolePolicies**
    - **ListRoles**
- Expand the **Resources** section and for all three resource types (*instance-profile*, *policy*, and *role*) select **Any in this account**.
- Back at the top of the screen, choose the **JSON** tab
  This view shows the JSON document that you just composed by using the visual editor.
- Verify that the policy document details match what is shown in the following example:

41. Exit the **Create policy** wizard by choosing **Cancel**.
42. In the **Policies** search box, search for LimitedIamPolicy.Observe that the policy details match the one you worked to build, as shown in the previous screen capture.
43. Edit the **DBAdministrators** IAM group.
    - **Attach** the **LimitedIamPolicy** policy
    - **Remove** the **IAMReadOnlyAccess** policy
44. Sofía asks Olivia to confirm that she can still access the details of the *CafeRole* IAM role, even with the more limited IAM access rights now granted to her.
45. Return to the browser tab where you are logged in as *Olivia*, and verify that you can still access the details of *CafeRole*.
    - In the **Amazon EC2** console, select the **aws-cloud9-DEVCafeServer** instance.
    - In the **Details** tab, notice that you can now see that the IAM role attached is named CafeRole.
    - Still as Olivia, go to the IAM console and choose **Roles**.
    - Search for and select the **CafeRole**.
    - In the Permissions tab, expand the **AmazonSSMManagedInstanceCore** policy and verify that you can still see the JSON document details.