Demo Project: Create EKS cluster with Fargate profile

This project builds on the earlier project, **Demo Project: Create AWS EKS Cluster with a Node Group**. Follow these instructions to create the necessary IAM role, Fargate profile, deploy a sample pod, and then clean up your resources.

Step 1: Create Fargate IAM Role

1. **Log in to the AWS Management Console.**
2. **Navigate to IAM Roles:**
   - Go to **IAM** → **Roles**.
3. **Create a New Role:**
   - Click on **Create role**.
4. **Select Trusted Entity:**
   - Choose **AWS Service**.
   - Under **Use case**, select **EKS**.
   - Then choose **EKS – Fargate pod**.
5. **Configure Role Details:**
   - **Role Name:** Enter `eks-fargate-role`.
6. **Create Role:**
   - Click **Create role** to finish.

Step 2: Create Fargate Profile

1. **Open the EKS Console:**
   - Navigate to **EKS** → **Clusters**.
2. **Select Your Cluster:**
   - Click on your cluster (for example, `eks-cluster-test`).
3. **Add a Fargate Profile:**
   - Click on **Add Fargate profile**.
4. **Configure Fargate Profile Settings:**
   - **Profile Name:** Enter `dev-profile`.
   - **Fargate IAM Role:** Confirm that it is set to `eks-fargate-role`.
   - **Subnets:** Verify that the correct private subnets are selected.
5. **Set Pod Selector:**
   - **Namespace:** Enter `dev` (this should match the namespace defined in your `nginx-config.yaml`).
   - **Match Labels:** Add a label with:
     - **Key:** `profile`
     - **Value:** `fargate` (ensure this matches the label in your `nginx-config.yaml`).
6. **Review and Create:**
   - Click **Next**, review your settings, and then click **Create**.

Step 3: Deploy Pod through Fargate

1. **Ensure the Namespace Exists:**
   - Create the `dev` namespace if it does not already exist: `kubectl create ns dev`
   - Verify the namespace: `kubectl get ns`
2. **Deploy Your Pod:**

- Apply your pod configuration (e.g., `nginx-config.yaml`): `kubectl apply -f nginx-config.yaml`

3. **Monitor the Pod Status:**
   - Watch the pod status in the `dev` namespace until they are running: `kubectl get pods -n dev -w`
4. **(Optional) Verify Node Details:**
   - Check the nodes (note that Fargate nodes are managed by AWS and might not appear like typical EC2 instances): `kubectl get nodes -n dev`

Step 4: Cleanup cluster resources

1. **Remove Cluster Components:**
   - First, remove any Node Groups and Fargate profiles from your EKS cluster.
2. **Delete the EKS Cluster:**
   - In the EKS console, click on the top right corner option **Delete Cluster**.
3. **Delete IAM Roles:**
   - Navigate to **IAM Roles** in the AWS Console.
   - Delete the following roles:
     - `eks-cluster-role`
     - `eks-fargate-role`
     - `eks-node-group-role`
4. **Confirm EC2 Instance Termination:**
   - Verify that any EC2 instances (if applicable) are automatically terminated.
5. **Clean Up CloudFormation Stacks:**
   - Go to **AWS CloudFormation**.
   - Select the stack associated with your EKS resources.
   - Click on **Delete** to remove any remaining resources.
