Here’s a **1-hour detailed, practical-oriented lesson plan** focused on **AWS IAM concepts for a DevOps course**, including hands-on labs, real-world scenarios, and troubleshooting exercises. This is structured for a live class but can be adapted for self-paced learning.

**AWS IAM Concepts for DevOps Engineers**

**Duration**: 60 minutes  
**Objective**: Teach students how to securely manage access to AWS resources using IAM, with a focus on DevOps use cases.

**1. Introduction to IAM (10 minutes)**

**Key Concepts**:

* **What is IAM?**
  + Identity and Access Management (IAM) controls who (users/roles) can do what (actions) on which resources.
  + Critical for enforcing the **principle of least privilege**.
* **Why IAM Matters in DevOps**:
  + Securing CI/CD pipelines.
  + Managing permissions for automated deployments (e.g., EC2, Lambda, CodePipeline).
  + Auditing access with CloudTrail.

**Demo**:

* Show the **IAM Dashboard** in the AWS Console.
* Highlight **Users**, **Groups**, **Roles**, and **Policies**.

**2. IAM Policies (15 minutes)**

**Key Concepts**:

* **Policy Structure**:
  + JSON-based documents with Effect, Action, Resource, and Condition.
  + Example:

json

Copy

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::my-devops-bucket/\*"

}

]

}

* **Policy Types**:
  + Managed Policies (AWS-managed or custom).
  + Inline Policies (attached directly to users/roles).

**Hands-On Lab**:

1. **Create a Custom Policy**:
   * Use the AWS Console to create a policy allowing read-only access to a specific S3 bucket.
2. **Attach Policy to a User**:
   * Create a test user and attach the policy.
3. **Test Permissions**:
   * Use the AWS CLI to test access:

bash

Copy

aws s3 ls s3://my-devops-bucket --profile test-user

**3. IAM Roles (15 minutes)**

**Key Concepts**:

* **Roles vs. Users**:
  + Roles are temporary credentials (no long-term access keys).
  + Used by AWS services (e.g., EC2, Lambda) or for cross-account access.
* **Trust Relationships**:
  + Define who (AWS service, account, or user) can assume the role.
  + Example Trust Policy for EC2:

json

Copy

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": { "Service": "ec2.amazonaws.com" },

"Action": "sts:AssumeRole"

}

]

}

**Hands-On Lab**:

1. **Create a Role for EC2**:
   * Create a role with AmazonS3ReadOnlyAccess and attach it to an EC2 instance.
2. **SSH into the EC2 Instance**:
   * Use the AWS CLI on the EC2 instance to list S3 buckets:

bash

Copy

aws s3 ls

1. **Troubleshooting Exercise**:
   * Intentionally misconfigure the role’s trust policy and have students debug access issues.

**4. IAM Best Practices for DevOps (10 minutes)**

**Key Concepts**:

* **Least Privilege**:
  + Grant only the permissions needed for the task.
* **Temporary Credentials**:
  + Use roles instead of long-term access keys.
* **IAM Groups**:
  + Group users by function (e.g., DevOps-Admins, Developers).
* **Auditing**:
  + Use **IAM Access Analyzer** to validate policies.
  + Monitor activity with **CloudTrail**.

**Demo**:

* Show how to rotate access keys and enable MFA.

**5. Advanced DevOps Scenario (10 minutes)**

**Use Case**: Cross-Account Access for CI/CD Pipeline

1. **Scenario**:
   * A CI/CD tool in Account A needs to deploy resources to Account B.
2. **Steps**:
   * Create a role in Account B with a trust policy allowing Account A.
   * Attach a deployment policy (e.g., AmazonEC2FullAccess).
   * Assume the role from Account A using AWS STS:

bash

Copy

aws sts assume-role --role-arn arn:aws:iam::ACCOUNT\_B\_ID:role/DeployRole --role-session-name CI-CD

1. **Lab**:
   * Students set up cross-account access and deploy a sample CloudFormation stack.

**6. Q&A and Recap (5 minutes)**

* Recap key concepts: Policies, Roles, Trust Relationships.
* Emphasize **least privilege** and **auditing**.
* Share resources:
  + [IAM Documentation](https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html)
  + [IAM Policy Simulator](https://policysim.aws.amazon.com/)

**Homework/Extended Lab:**

* **Task**: Create a CI/CD pipeline using CodePipeline that deploys to an S3 bucket.
  + Use IAM roles for CodePipeline and CodeBuild.
  + Restrict the role to only the required S3 bucket.