**Resolving Persistent Volume Claim (PVC) Pending Issues with AWS StorageClass (gp2)**

When working with Kubernetes clusters on AWS, you may encounter issues where Persistent Volume Claims (PVCs) remain in a Pending state. This is often caused by misconfigurations in the associated StorageClass. This guide documents the steps to resolve this issue.

**Problem**

PVCs using the gp2 StorageClass were stuck in a Pending state, with errors indicating:

* Missing permissions to perform ec2:CreateVolume.
* Incorrect volumeBindingMode settings in the gp2 StorageClass.

**Example Error**

text

Copy code

failed to provision volume with StorageClass "gp2": rpc error: code = Internal desc = Could not create volume in EC2: operation error EC2: CreateVolume, UnauthorizedOperation

**Solution**

**Step 1: Verify StorageClass Configuration**

1. **Check the current configuration of the gp2 StorageClass:**

bash

Copy code

kubectl describe storageclass gp2

Example output:

text

Copy code

Provisioner: kubernetes.io/aws-ebs

Parameters: fsType=ext4,type=gp2

VolumeBindingMode: WaitForFirstConsumer

**Key observation:** VolumeBindingMode is set to WaitForFirstConsumer, which delays volume creation until the pod is scheduled. This mode may cause delays if the cluster node IAM roles lack proper permissions.

**Step 2: Set Default StorageClass**

To ensure new PVCs automatically use the gp2 StorageClass:

1. **Patch the gp2 StorageClass to make it the default:**

bash

Copy code

kubectl patch storageclass gp2 -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'

1. Verify the update:

bash

Copy code

kubectl describe storageclass gp2

Check that the annotation storageclass.kubernetes.io/is-default-class: "true" is present.

**Step 3: Add IAM Permissions to Allow EC2 Volume Creation**

This step ensures the EKS worker node role has the required permissions to create and manage EBS volumes.

**Using the AWS Management Console (GUI)**

1. **Log in to the AWS Management Console.**
2. Navigate to the **IAM** section from the dashboard.
3. **Find the EKS Node Role:**
   * Click on **Roles** in the left-hand menu.
   * Search for the role associated with your EKS worker nodes (e.g., SUE\_Node\_Group\_Role).
   * Click on the role name to open its details.
4. **Attach a New Policy:**
   * On the role's page, go to the **Permissions** tab.
   * Click on **Add permissions** → **Attach policies**.
   * Search for AmazonEC2FullAccess in the policy list.
   * Select the policy and click **Attach policies**.
5. **(Optional)** Create a Custom Policy (if full access is not desired):
   * Instead of using AmazonEC2FullAccess, you can create a custom policy:
     + Go to the **Policies** section in IAM.
     + Click **Create policy**.
     + Use the following JSON policy:

json

Copy code

{

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

"Statement": [

{

"Effect": "Allow",

"Action": [

"ec2:CreateVolume",

"ec2:AttachVolume",

"ec2:DeleteVolume",

"ec2:DescribeVolumes",

"ec2:DescribeVolumeStatus"

],

"Resource": "\*"

}

]

}

* + - Name the policy (e.g., EKS-VolumePermissions) and save it.
    - Go back to the **Roles** page, find your EKS worker node role, and attach this custom policy.

**Using the AWS CLI**

1. **Attach the Managed Policy:**

To give full EC2 permissions, use the following command:

bash

Copy code

aws iam attach-role-policy \

--role-name SUE\_Node\_Group\_Role \

--policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess

1. **Attach a Custom Policy (if full access is not desired):**

Create a policy with only the required actions:

json

Copy code

{

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

"Statement": [

{

"Effect": "Allow",

"Action": [

"ec2:CreateVolume",

"ec2:AttachVolume",

"ec2:DeleteVolume",

"ec2:DescribeVolumes",

"ec2:DescribeVolumeStatus"

],

"Resource": "\*"

}

]

}

* + Save this JSON to a file named eks-volume-policy.json.
  + Create the policy:

bash

Copy code

aws iam create-policy --policy-name EKS-VolumePermissions --policy-document file://eks-volume-policy.json

* + Attach it to the role:

bash

Copy code

aws iam attach-role-policy \

--role-name SUE\_Node\_Group\_Role \

--policy-arn arn:aws:iam::<your-account-id>:policy/EKS-VolumePermissions

**Step 4: Update the volumeBindingMode (Optional)**

By default, the gp2 StorageClass uses WaitForFirstConsumer. You can change this to Immediate to provision volumes as soon as the PVC is created:

1. **Patch the gp2 StorageClass:**

bash

Copy code

kubectl patch storageclass gp2 --type=merge -p '{"volumeBindingMode":"Immediate"}'

1. Verify the update:

bash

Copy code

kubectl describe storageclass gp2

Confirm VolumeBindingMode: Immediate.

**Step 5: Retry PVC Creation**

1. Delete any stuck PVCs:

bash

Copy code

kubectl delete pvc <pvc-name> -n <namespace>

1. Recreate your PVCs and ensure they bind successfully to the provisioned volumes.
2. Verify the PVC status:

bash

Copy code

kubectl get pvc -n <namespace>

Output should show the PVC in Bound state.