📋 AWS REKOGNITION SETUP: COMPLETE STEP-BY-STEP WALKTHROUGH

🎯 OBJECTIVE

Transform the Core Safety Loop system from demo mode to full production functionality by configuring AWS IAM permissions and setting up Face Collections.

📊 CURRENT STATUS VERIFICATION

Step 0: Check Current System Status

```
cd /home/ubuntu/safeplay-staging
node test-core-safety-loop.js
```

Expected Output (Demo Mode):

```
🎯 Testing Core Safety Loop System
✅ Core files present
⚠️  AWS permissions needed: AccessDeniedException
📋 System Status Summary:
   • Rekognition: ❌ Permissions needed
   • Demo Mode: ✅ Active (AWS permissions needed)
```

🔐 PART 1: AWS IAM PERMISSIONS SETUP

Step 1.1: Access AWS Management Console

1. Log into your AWS Management Console
2. Ensure you're in the correct region: **us-east-1**
3. Navigate to **IAM** (Identity and Access Management)

Step 1.2: Locate Your IAM User/Role

1. Click on **"Users"** in the left sidebar
2. Look for user: **spark-permissions** (or the user associated with your credentials)
3. **Alternative**: Check **"Roles"** if you're using role-based access

Step 1.3: Create SafePlay Rekognition Policy

1. In IAM, click **"Policies"** in the left sidebar
2. Click **"Create policy"**
3. Select the **"JSON"** tab
4. **Copy and paste this exact policy:**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RekognitionFaceCollections",
            "Effect": "Allow",
            "Action": [
                "rekognition:CreateCollection",
                "rekognition:DeleteCollection",
                "rekognition:ListCollections",
                "rekognition:DescribeCollection",
                "rekognition:IndexFaces",
                "rekognition:SearchFacesByImage",
                "rekognition:SearchFaces",
                "rekognition:DeleteFaces",
                "rekognition:ListFaces",
                "rekognition:DetectFaces",
                "rekognition:CompareFaces",
                "rekognition:DetectModerationLabels"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3FaceStorage",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::safeplay-faces",
                "arn:aws:s3:::safeplay-faces/*"
            ]
        }
    ]
}
```

1. Click **“Next: Tags”** (skip tags for now)
2. Click **“Next: Review”**
3. **Name**: `SafePlayRekognitionPolicy`
4. **Description**: `Permissions for SafePlay Core Safety Loop face recognition system`
5. Click **“Create policy”**

Step 1.4: Attach Policy to Your User/Role

1. Go back to **“Users”** (or “Roles”) in IAM
2. Click on your user: **spark-permissions**
3. Click the **“Permissions”** tab
4. Click **“Add permissions”** → **“Attach existing policies directly”**
5. Search for: `SafePlayRekognitionPolicy`
6. Check the box next to it
7. Click **“Next: Review”** → **“Add permissions”**

Step 1.5: Verify Policy Attachment

1. In your user's Permissions tab, confirm you see:
   - `SafePlayRekognitionPolicy` ✅
2. **Click on the policy name** to verify it shows the JSON you pasted

---

🧪 PART 2: IMMEDIATE VERIFICATION

Step 2.1: Test AWS Connection

```
cd /home/ubuntu/safeplay-staging
node test-core-safety-loop.js
```

Expected Output (Success):

```
✅ AWS credentials working
   Collections found: 0
✅ AWS Configuration:
   • Credentials: ✅ Valid
   • Rekognition: ✅ Connected
```

If you still see permissions errors:
- Wait 2-3 minutes for AWS IAM changes to propagate
- Try running the test again
- Verify you attached the policy correctly

Step 2.2: Direct Permission Test

```
cd /home/ubuntu/safeplay-staging
node -e "
const { RekognitionClient, ListCollectionsCommand } = require('@aws-sdk/client-rekogni-
tion');

const client = new RekognitionClient({
  region: process.env.AWS_REGION || 'us-east-1',
  credentials: {
    accessKeyId: process.env.AWS_ACCESS_KEY_ID,
    secretAccessKey: process.env.AWS_SECRET_ACCESS_KEY,
    sessionToken: process.env.AWS_SESSION_TOKEN
  }
});

(async () => {
  try {
    const result = await client.send(new ListCollectionsCommand({}));
    console.log('✅ SUCCESS: Rekognition permissions working!');
    console.log('Collections found:', result.CollectionIds?.length || 0);
    console.log('Available collections:', result.CollectionIds || []);
  } catch (error) {
    console.log('❌ FAILED: Rekognition permissions still needed');
    console.log('Error:', error.name, '-', error.message);
  }
})();
"
```

🏗️ PART 3: FACE COLLECTIONS SETUP

Step 3.1: Initialize Face Collections

```
cd /home/ubuntu/safeplay-staging
node scripts/setup-face-collections.js
```

Expected Output:

```
🔐 Setting up Face Collections for SafePlay Venues
✅ AWS Rekognition connected successfully!
Existing collections: 0

Found X venues
✅ Created collection: safeplay-venue-[venue-id] (Demo Venue)
✅ Created demo collection: safeplay-demo-main

📊 Summary:
Collections created: X
Total venues: X
✅ Face collection setup completed!
```

Step 3.2: Verify Collections via API

Start the development server:

```
cd /home/ubuntu/safeplay-staging
npm run dev
```

Test the collections API:

```
# In a new terminal window
curl "http://localhost:3000/api/faces/collections" \
  -H "Cookie: next-auth.session-token=your-session-token"
```

Or test via browser:
1. Open: `http://localhost:3000/venue-admin/core-safety-loop`
2. Click the **“Hardware”** tab
3. Look for face collection information

Step 3.3: Check System Status

```
curl "http://localhost:3000/api/system/aws-status" \
  -H "Cookie: next-auth.session-token=your-session-token"
```

Expected Response:

```
{
  "success": true,
  "systemHealth": {
    "awsCredentials": true,
    "rekognitionPermissions": true,
    "faceCollections": true,
    "overallStatus": "operational"
  },
  "recommendations": []
}
```

👶 PART 4: FACE ENROLLMENT TESTING

Step 4.1: Access Core Safety Loop Interface

1. Open browser: `http://localhost:3000/venue-admin/core-safety-loop`
2. **Login as venue admin** or use demo credentials
3. Navigate to **“Hardware”** tab

Step 4.2: Test Face Detection

1. In the Hardware tab, find **“Face Recognition Setup”**
2. Click **“Test Face Detection”**
3. Upload a clear photo of a face
4. **Expected result**: Face detected with confidence score

Step 4.3: Enroll a Child’s Face

1. Go to venue children management
2. Select a child

3. Click **“Enroll Face”**
4. Upload a clear, well-lit photo showing only the child’s face
5. **Success criteria**:
   - Face detected: ✅
   - Confidence > 90%: ✅
   - Single face only: ✅
   - Enrollment successful: ✅

Step 4.4: Test Face Recognition

1. In Core Safety Loop → **“Hardware”** tab
2. Click **“Test Recognition”**
3. Upload a different photo of the same child
4. **Expected result**: Child identified with match confidence

---

📊 PART 5: REAL-TIME SYSTEM VERIFICATION

Step 5.1: Live Tracking Dashboard

1. Core Safety Loop → **“Live Tracking”** tab
2. **Verify you see**:
   - Zone map with children locations
   - Real-time activity feed
   - Child safety status indicators
   - **“Demo Mode: OFF”** indicator

Step 5.2: Camera Feeds

1. Core Safety Loop → **“Camera Feeds”** tab
2. **Verify**:
   - Camera discovery working
   - Face recognition overlay enabled
   - Real-time detection events

Step 5.3: Final System Test

```
cd /home/ubuntu/safeplay-staging
node test-core-safety-loop.js
```

Expected Final Output:

```
✅ AWS credentials working
   Collections found: X
📋 System Status Summary:
🔐 AWS Configuration:
   • Credentials: ✅ Valid
   • Rekognition: ✅ Connected
🎯 Core Safety Loop:
   • Real-time Face Recognition: ✅ Implemented
   • Live Tracking Service: ✅ Implemented
   • Camera Hardware Integration: ✅ Implemented
📊 Face Recognition:
   • Demo Mode: ❌ Disabled
🚀 Next Steps:
   • Access Core Safety Loop at: /venue-admin/core-safety-loop
```

🚨 TROUBLESHOOTING GUIDE

Problem: AWS Connection Still Fails

Check 1: Credential Source

```
echo "Access Key: ${AWS_ACCESS_KEY_ID:0:8}..."
echo "Secret Key: ${AWS_SECRET_ACCESS_KEY:0:8}..."
echo "Region: $AWS_REGION"
```

Check 2: IAM Policy Attachment
- AWS Console → IAM → Users → [your-user] → Permissions
- Verify `SafePlayRekognitionPolicy` is listed
- Click policy name → verify JSON content

Check 3: Permission Propagation
- Wait 5 minutes after attaching policy
- Try logging out and back into AWS Console
- Re-run test script

Problem: Face Collections Fail to Create

Error: `ResourceAlreadyExistsException`
- Collections already exist (this is OK)
- Run: `node -e "console.log('Collections exist, proceeding...')"`

Error: `ValidationException`
- Check collection naming (alphanumeric only)
- Verify venue IDs are valid UUIDs

Problem: Face Enrollment Fails

“No face detected”:
- Use well-lit, clear photos
- Ensure face is clearly visible
- Face should be looking toward camera
- No sunglasses or face coverings

“Multiple faces detected”:
- Crop photo to show only one person
- Ensure background doesn’t contain faces

“Low confidence”:
- Improve photo quality
- Use higher resolution image
- Ensure good lighting

Problem: Demo Mode Still Active

Check API Status:

```
curl "http://localhost:3000/api/system/aws-status"
```

Verify Response:
- `systemHealth.rekognitionPermissions: true`
- `faceRecognition.demoMode: false`

✅ SUCCESS VERIFICATION CHECKLIST

AWS Configuration:

- [ ] IAM policy created: `SafePlayRekognitionPolicy`
- [ ] Policy attached to user/role
- [ ] AWS connection test passes
- [ ] `ListCollections` permission works

Face Collections:

- [ ] Collections created for all venues
- [ ] Demo collections created
- [ ] Database updated with collection IDs
- [ ] API endpoints responding correctly

Face Recognition:

- [ ] Face detection working
- [ ] Face enrollment successful
- [ ] Face recognition/matching working
- [ ] Confidence scores reasonable (>80%)

System Integration:

- [ ] Core Safety Loop interface accessible
- [ ] Live tracking functional
- [ ] Camera feeds working
- [ ] Demo mode disabled
- [ ] Real-time events broadcasting

Production Readiness:

- [ ] All API endpoints responding
- [ ] Error handling working correctly
- [ ] System status API shows “operational”
- [ ] No critical recommendations in status

🚀 NEXT STEPS AFTER SUCCESSFUL SETUP

Immediate Actions:

1. **Test with Real Children**: Enroll faces of actual children at your venue
2. **Configure Cameras**: Set up physical cameras and connect them
3. **Train Staff**: Show venue staff how to use the Core Safety Loop interface
4. **Set Alert Rules**: Configure safety alerts and notifications

Production Configuration:

1. **Backup Strategy**: Set up collection backups
2. **Monitoring**: Configure AWS CloudWatch alerts
3. **Performance Tuning**: Adjust confidence thresholds based on testing
4. **Security Review**: Ensure all face data is properly secured

System Optimization:

1. **Confidence Thresholds**: Fine-tune based on real-world performance
2. **Camera Placement**: Optimize camera positions for best recognition
3. **Zone Configuration**: Set up safety zones and boundaries
4. **Alert Customization**: Configure venue-specific alert rules

---

📞 SUPPORT AND RESOURCES

Documentation Files:

- `AWS_REKOGNITION_SETUP_GUIDE.md` - Original setup guide
- `test-core-safety-loop.js` - System testing script
- `scripts/setup-face-collections.js` - Collections setup

API Endpoints for Testing:

- `/api/system/aws-status` - Complete system status
- `/api/faces/collections` - Face collections management
- `/api/faces/enroll` - Face enrollment
- `/api/faces/test-recognition` - Recognition testing

User Interfaces:

- `/venue-admin/core-safety-loop` - Main Core Safety Loop interface
- `/venue-admin/core-safety-loop?tab=hardware` - Hardware configuration
- `/venue-admin/core-safety-loop?tab=cameras` - Camera feeds

Emergency Contacts:

- **AWS Support**: If you encounter AWS service issues
- **IAM Documentation**: https://docs.aws.amazon.com/iam/
- **Rekognition Docs**: https://docs.aws.amazon.com/rekognition/

🔄 QUICK VERIFICATION COMMANDS

Test everything at once:

```
cd /home/ubuntu/safeplay-staging

# 1. Test AWS connection
echo "Testing AWS connection..."
node test-core-safety-loop.js

# 2. Test API endpoints
echo "Testing API endpoints..."
curl -s "http://localhost:3000/api/system/aws-status" | jq '.system-
Health.overallStatus'

# 3. List collections
echo "Listing face collections..."
node -e "
const { RekognitionClient, ListCollectionsCommand } = require('@aws-sdk/client-rekogni-
tion');
const client = new RekognitionClient({
  region: process.env.AWS_REGION || 'us-east-1',
  credentials: {
    accessKeyId: process.env.AWS_ACCESS_KEY_ID,
    secretAccessKey: process.env.AWS_SECRET_ACCESS_KEY,
    sessionToken: process.env.AWS_SESSION_TOKEN
  }
});
client.send(new ListCollectionsCommand({}))
  .then(result => console.log('Collections:', result.CollectionIds))
  .catch(error => console.log('Error:', error.message));
"
```

Expected Success Output:

```
Testing AWS connection...
✅ Core Safety Loop system test completed!

Testing API endpoints...
"operational"

Listing face collections...
Collections: ["safeplay-venue-...", "safeplay-demo-main"]
```

🎉 CONGRATULATIONS!

If all steps completed successfully, your Core Safety Loop system is now **fully operational** with:

✅ **AWS Rekognition Integration**: Real face recognition
✅ **Face Collections**: Venue-specific child enrollment
✅ **Real-Time Tracking**: Live child location monitoring
✅ **Camera Integration**: Multi-vendor camera support
✅ **Production Ready**: Full functionality enabled

Your SafePlay Core Safety Loop is now protecting children with cutting-edge AI technology!