Agent Core IAM Role CDK Stack

This CDK stack creates an IAM role for AWS Bedrock Agent Core with all the necessary permissions.

Overview

The stack creates a secure IAM role named `AgentCore-ExecutionRole` that can be assumed by the Bedrock Agent Core service. This role includes comprehensive permissions for ECR access, CloudWatch logging, X-Ray tracing, and Bedrock model invocation.

Prerequisites

Before deploying this stack, ensure you have:

- **AWS CLI**: Installed and configured with appropriate credentials
- **AWS CDK CLI**: Version 2 (`npm install -g aws-cdk`)
- **Python**: Version 3.12 or later
- **Node.js**: Version 20.x or later (required for CDK CLI)
- **AWS Account**: Bootstrapped for CDK and With sufficient permissions to create IAM roles and policies
- **Python virtual environment**: With the dependencies installed

Verify Prerequisites

```
# Check AWS CLI configuration
aws sts get-caller-identity

# Check CDK CLI version
cdk --version

# Check Python version
python --version
```

Deployment

Quick Deploy (Default Configuration)

```
cdk deploy
```

This creates the IAM role with the default name `AgentCore-ExecutionRole`.

What Gets Created

The stack creates the following AWS resources:

IAM Role: `AgentCore-ExecutionRole`

Trust Policy: Allows assumption by `bedrock-agentcore.amazonaws.com` service with account and source ARN conditions.

Attached Policies:

1. **ECR Image Access**
   - `ecr:BatchGetImage`
   - `ecr:GetDownloadUrlForLayer`
   - Resources: All ECR repositories in your account
2. **CloudWatch Logs Access**
   - `logs:DescribeLogStreams`
   - `logs:CreateLogGroup`
   - `logs:DescribeLogGroups`
   - `logs:CreateLogStream`
   - `logs:PutLogEvents`
   - Resources: Bedrock Agent Core log groups
3. **ECR Authorization**
   - `ecr:GetAuthorizationToken`
   - Resources: All (required for ECR access)
4. **X-Ray Tracing**
   - `xray:PutTraceSegments`
   - `xray:PutTelemetryRecords`
   - `xray:GetSamplingRules`
   - `xray:GetSamplingTargets`
   - Resources: All
5. **CloudWatch Metrics**
   - `cloudwatch:PutMetricData`
   - Namespace: `bedrock-agentcore`
6. **Agent Access Tokens**
   - `bedrock-agentcore:GetWorkloadAccessToken`
   - `bedrock-agentcore:GetWorkloadAccessTokenForJWT`
   - `bedrock-agentcore:GetWorkloadAccessTokenForUserId`
   - Resources: Workload identity directories
7. **Bedrock Model Invocation**
   - `bedrock:InvokeModel`
   - `bedrock:InvokeModelWithResponseStream`
   - Resources: All foundation models and custom models

Stack Outputs

After successful deployment, the stack outputs:

- **AgentCoreRoleArn**: The ARN of the created IAM role

Export Role ARN to Environment Variable

For easy reference in other applications and scripts, export the role ARN to an environment variable:

```
# Export the role ARN to environment variable
export AGENT_CORE_ROLE_ARN=$(aws cloudformation describe-stacks \
  --stack-name AgentCoreIAMStack \
  --query 'Stacks[0].Outputs[?OutputKey==`AgentCoreRoleArn`].OutputValue'
\
  --output text)

# Verify the export
echo $AGENT_CORE_ROLE_ARN
```

Use in Other Applications

You can now reference the role ARN in your applications:

```
# Example: Use in AWS CLI commands
aws bedrock create-agent --agent-name MyAgent --execution-role-arn
$AGENT_CORE_ROLE_ARN

# Example: Use in Python applications
import os
role_arn = os.environ.get('AGENT_CORE_ROLE_ARN')
```

Verification

Verify Role Creation

```
# List the created role
aws iam get-role --role-name AgentCore-ExecutionRole

# List attached policies
aws iam list-attached-role-policies --role-name AgentCore-ExecutionRole
```

Troubleshooting

Common Issues

1. **CDK Bootstrap Required**

   ```
   Error: This stack uses assets, so the toolkit stack must be deployed
   ```

   Solution: Run `cdk bootstrap`

2. **Insufficient Permissions**

   ```
   Error: User is not authorized to perform: iam:CreateRole
   ```

   Solution: Ensure your AWS credentials have IAM permissions

3. **Stack Already Exists**

   ```
   Error: Stack AgentCoreIAMStack already exists
   ```

   Solution: Use `cdk deploy --force` or delete the existing stack first

Cleanup

Delete the Stack

```
cdk destroy
```

Security Considerations

- The role uses least-privilege principles where possible
- Trust policy restricts assumption to Bedrock Agent Core service only
- Source account and ARN conditions prevent cross-account access
- Some wildcard permissions are necessary for service functionality
- CDK-nag suppressions are applied for prototype/development use

Support

For issues related to:

- **CDK**: Check AWS CDK Documentation
- **Bedrock Agent Core**: Check AWS Bedrock Documentation
- **IAM**: Check AWS IAM Documentation