AWS CodeCommit

AWS CodeCommit

- AWS CodeCommit is a fully-managed source control service that hosts secure Git-based repositories. It makes it easy for teams to collaborate on code in a secure and highly scalable ecosystem
- CodeCommit eliminates the need to operate your own source control system or worry about scaling its infrastructure.
- You can use CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.

AWS CodeCommit - Features

- You can transfer your files to and from AWS CodeCommit using HTTPS or SSH, as you prefer. Your repositories are also automatically encrypted at rest through AWS Key Management Service (AWS KMS) using customer-specific keys.
- AWS CodeCommit uses AWS Identity and Access Management to control and monitor who can access your data as well as how, when, and where they can access it. CodeCommit also helps you monitor your repositories via AWS CloudTrail and AWS CloudWatch.

AWS CodeCommit - Features

- AWS CodeCommit stores your repositories in Amazon S3 and Amazon DynamoDB. Your encrypted data is redundantly stored across multiple facilities. This architecture increases the availability and durability of your repository data.
- AWS CodeCommit allows you to create as many repositories as you need, with up to 1,000 repositories by default and no limits upon request.

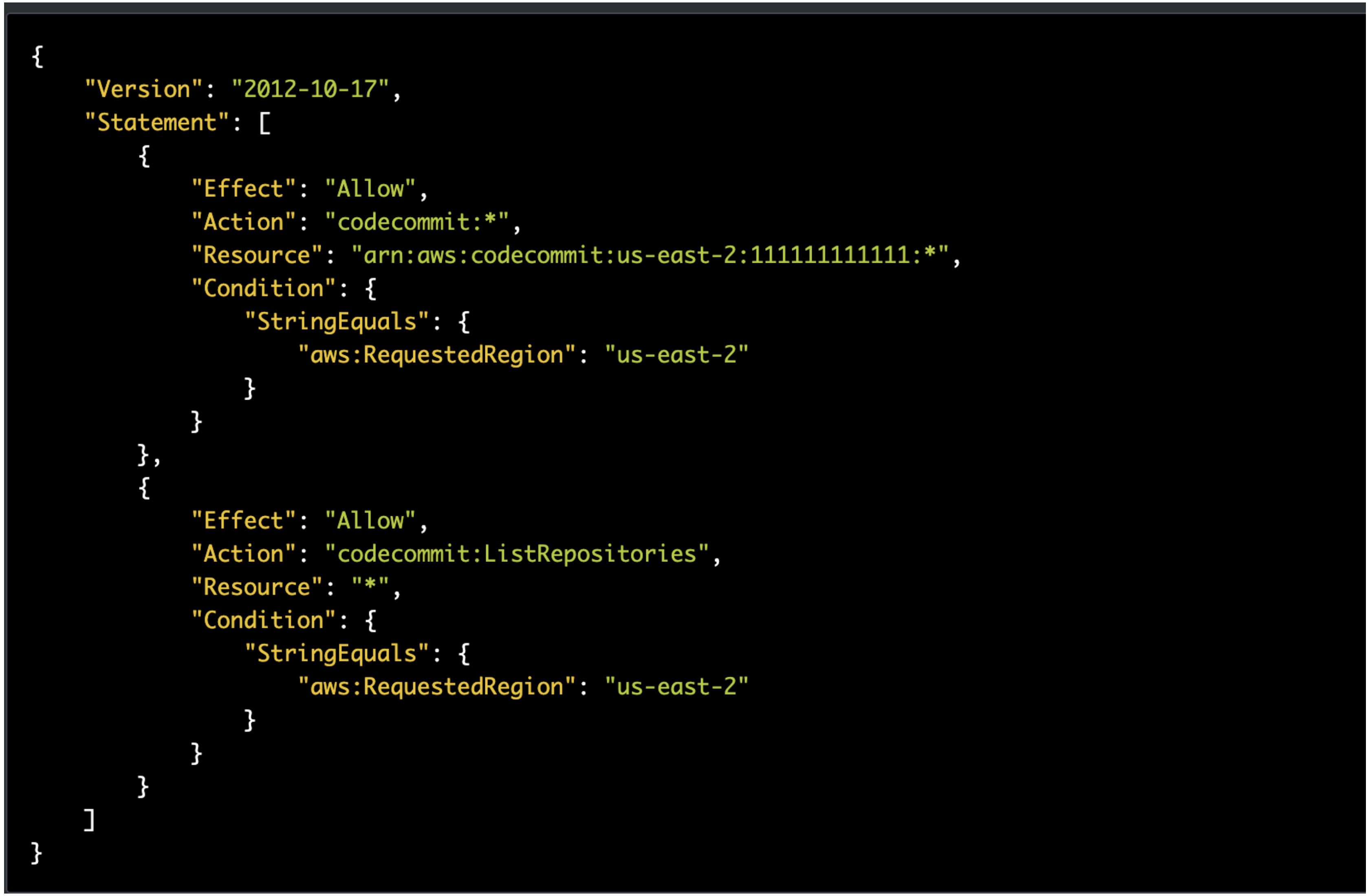
AWS CodeCommit - Features

- You can now receive notifications for events impacting your repositories. Notifications will come in the form of Amazon SNS notifications. Each notification will include a status message as well as a link to the resources whose event generated that notification. Additionally, using AWS CodeCommit repository triggers, you can send notifications and create HTTP webhooks with Amazon SNS or invoke AWS Lambda functions in response to the repository events you choose.

AWS CodeCommit – CloudWatch Events

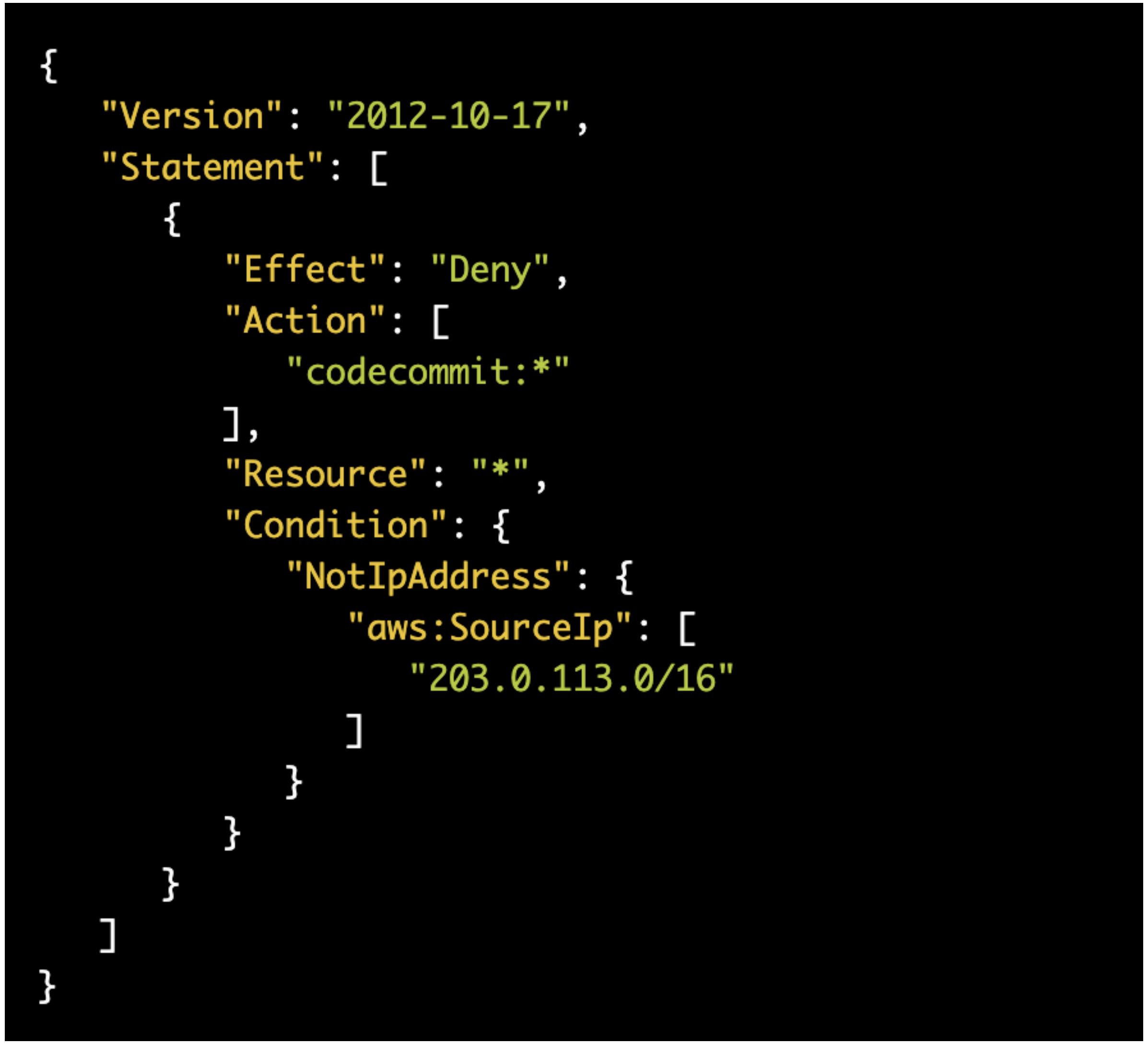
- AWS CodeCommit now sends repository state changes to Amazon CloudWatch events. For example, the repository state changes include activities like pushing new code to a code repository. Using these new event types, customers can build Amazon CloudWatch Event rules to match AWS CodeCommit events and route them to one or more targets like an Amazon SNS Topic, AWS Step Functions state machine, or AWS Lambda function to trigger automated workflows to process repository changes.

AWS CodeCommit – IAM Policies



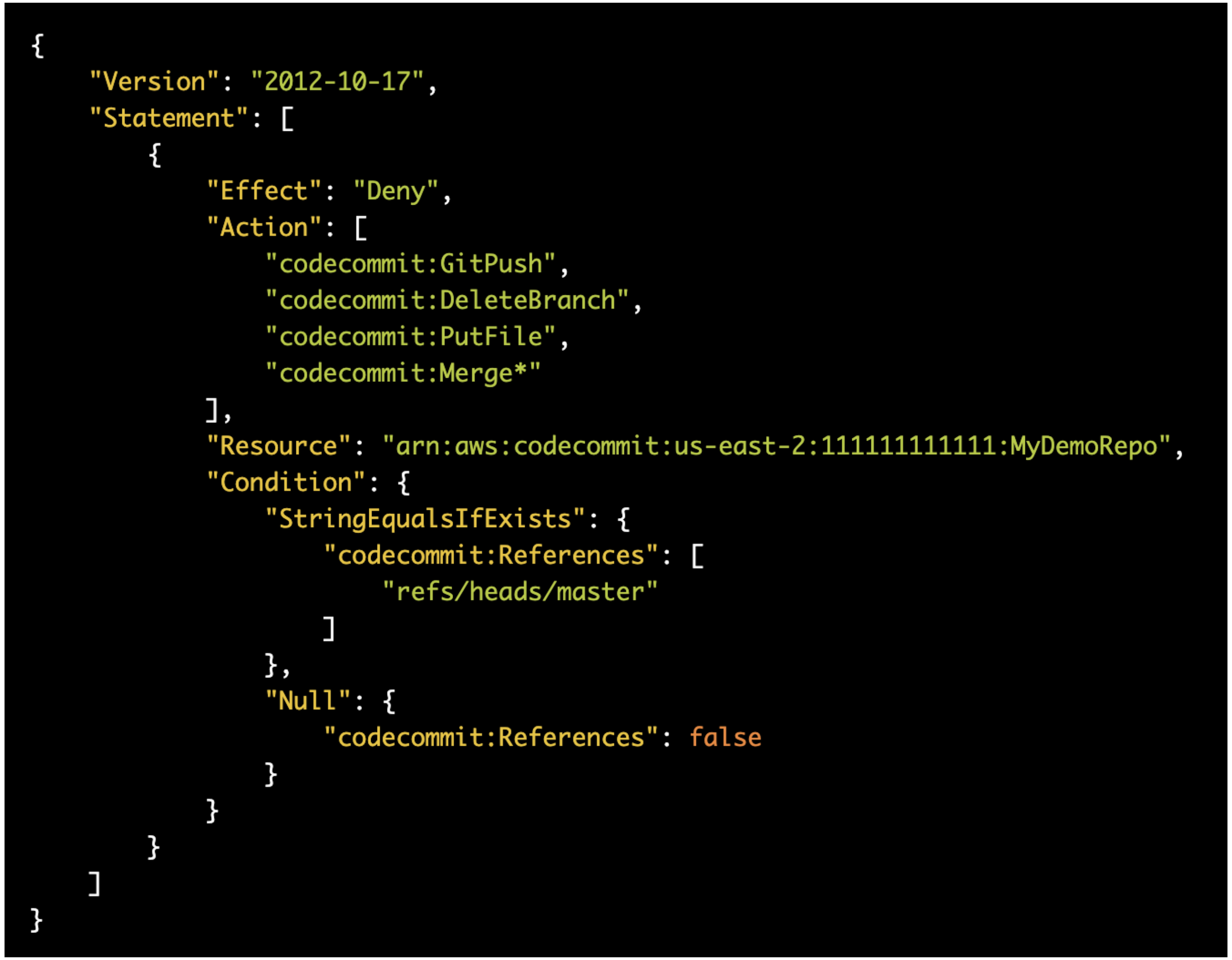
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "codecommit:*",
            "Resource": "arn:aws:codecommit:us-east-2:111111111111:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": "us-east-2"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "codecommit:ListRepositories",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": "us-east-2"
                }
            }
        }
    ]
}
```

AWS CodeCommit – IAM Policies



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "codecommit:*"
            ],
            "Resource": "*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "203.0.113.0/16"
                    ]
                }
            }
        }
    ]
}
```

AWS CodeCommit – IAM Policies



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "codecommit:GitPush",
                "codecommit:DeleteBranch",
                "codecommit:PutFile",
                "codecommit:Merge*"
            ],
            "Resource": "arn:aws:codecommit:us-east-2:111111111111:MyDemoRepo",
            "Condition": {
                "StringEqualsIfExists": {
                    "codecommit:References": [
                        "refs/heads/master"
                    ]
                },
                "Null": {
                    "codecommit:References": false
                }
            }
        }
    ]
}
```

A Deny policy that denies users the ability to make changes to a branch named master, including deleting that branch, in a repository named MyDemoRepo. You can use this policy with the AWSCodeCommitPowerUser managed policy. Users with these two policies applied would be able to create and delete branches, create pull requests, and all other actions as allowed by AWSCodeCommitPowerUser, but they would not be able to push changes to the branch named master.