Configure and Work with CodeCommit from the CLI

Introduction

In this hands-on lab, we go through the process of configuring a CodeCommit repository from the AWS Command Line Interface. We also go through some key prerequisite steps, such as installing Git, configuring HTTPS credentials, and preparing the environment to work with Git and CodeCommit. This lab covers the steps that need to be taken anytime you work with CodeCommit from the CLI.

Solution

Make sure you're in the N. Virginia (`us-east-1`) region throughout the lab.

Create a Server from an Amazon Linux 2 AMI

1. Navigate to the EC2 dashboard, and click Launch Instance.
2. On the AMI page, select the Amazon Linux 2 AMI.
3. Leave *t2.micro* selected, and click Next: Configure Instance Details.
4. On the *Configure Instance Details* page:
    - *Network*: Leave default
    - *Subnet*: Select one provided
    - *Auto-assign Public IP*: Enable
5. Click Review and Launch > Launch.
6. In the key pair dialog, select Create a new key pair.
7. Give it a *Key pair name* of "confighttps".
8. Click Download Key Pair, and then Launch Instances.
9. Click View Instances, and give it a few minutes to enter the *running* state.

Install Git and Create IAM CodeCommit Credentials

1. Select the newly created instance.
2. Click Connect at the top of the screen.
3. Open a terminal window via the instance connect option.
4. However you can connect via whatever terminal connection you like.
5. Linux/Mac users: Execute the following command:
6. `chmod 400 confighttps.pem`

7. Linux/Mac users: Run the example SSH command from the *Connect* window in the AWS Management Console in your terminal application.
   Note: If you are using Windows/PuTTY, use the link in the *Connect* window in the AWS Management Console for *Connect using PuTTY* and follow those instructions to connect.
8. In the terminal, run the following command:
9. `sudo yum update -y`
10. In the AWS Management Console, navigate to IAM > Users.
11. Click on add user and name whatever you want.
12. Give both Access types and create a custom password.
13. Uncheck Require password reset.
14. Click Next permissions > Attach existing policies directly.
15. Type "AWSCodeCommit" in the filter search box.
16. Select the `AWSCodeCommitFullAccess` policy.
17. Click Next:Tags, Next:Review and Create User
18. Download .csv and close.
    In the terminal, run the following command to install Git:
19. `sudo yum install git -y`
20. In the AWS Management Console, with `new user` opened, click the Security credentials tab.
21. Navigate to the bottom of the screen, and click Generate under *HTTPS Git credentials for AWS CodeCommit*.
22. Click Download credentials > Close.

Create IAM User Access Key and CodeCommit Repository

1. *Note:* The access keys will be used for the AWS Command Line Interface access in our lab. Generally, you would use an IAM role attached to the instance in production environments. The HTTPS Git credentials for AWS CodeCommit will provide access to CodeCommit.
2. Open the terminal window.
3. Configure the AWS CLI credentials:
4. `aws configure`
5. Enter the AWS `Access Key` and `Secret Access Key` from the file downloaded in the previous step when prompted to do so.
6. Enter `us-east-1` as the region.
7. Hit Enter to accept the `Default output format`.
8. Run the following command to create a CodeCommit repository from the AWS CLI:

9. `aws codecommit create-repository --repository-name RepoFromCLI --repository-description "My demonstration repository"`
10. In the AWS Management Console, open CodeCommit and refresh the screen.
11. Click on the RepoFromCLI repository link to open it.
12. Click Add file > Upload file.
13. Click Choose file.
14. Select any small file from your machine you don't mind uploading.
15. Enter your name as the author name and your email as the email address.
16. Click Commit changes.
17. Click Clone URL > Clone HTTPS.
18. Open up your terminal window.
19. Run the following command to clone the repository to the server:
20. `git clone [paste the URL from the clipboard]`
21. Enter the Git credentials username from the file downloaded earlier for the username prompt.
22. Enter the Git credentials password from the file downloaded earlier for the password prompt.
23. Run the following commands:

```
ls
cd RepoFromCLI
```

24. `ls`
25. Run the following command to create a local text file:
26. `vim test.txt`
27. Hit `i` to enter insert mode and type the text:
28. `This is just a test of working with CodeCommit from the CLI`
29. Hit the Escape key, type `:wq!`, and press Enter.
30. Add the file to Git:
31. `git add test.txt`
32. Commit the file to the local repository:
33. `git commit -m "added test.txt"`
34. Verify the file was committed:
35. `git log`
36. Push the change to the CodeCommit repository:
37. `git push -u origin main`

38. Enter the Git credentials username from the file downloaded earlier for the username prompt.
39. Enter the Git credentials password from the file downloaded earlier for the password prompt.
40. Refresh the AWS Management Console view of the CodeCommit repository for *RepoFromCLI* and verify the new file was uploaded.

Conclusion

Congratulations on successfully completing this hands-on lab!