**Monitoring AWS CodePipeline Changes Through AWS CloudWatch Events Rules**

**Introduction**

Your company decided to automate its application deployment process with AWS CodePipeline. The management team would like the DevOps team to closely monitor the pipeline by triggering email notifications for any events that occur in the pipeline. In this lab, we will set up CloudWatch events to monitor a CodePipeline. By creating an SNS topic, we can link our CloudWatch event to the SNS topic. Subscribers to the topic will then be notified via email of changes in our CodePipeline.

**Solution**

Log in to the live AWS environment using the credentials provided. Make sure you're in the N. Virginia (us-east-1) region throughout the lab.

Download the MessageUtil.zip file provided on the [lab GitHub page](https://github.com/natonic/Developer-Tools-Deep-Dive/tree/master/Labs/MonitorCodepipelineThruCloudWatch) as we need to upload it for a lab task.

**Set Up Amazon SNS Topic and Email Subscription**

1. Navigate to SNS.
2. Click **Create topic**.
3. Give it a *Topic name* of "pipelineactivity".
4. Leave the other defaults for now, and click **Create topic**.
5. Once it's created, click **Create subscription**.
6. On the *Create subscription* page, set the following values:
   * *Protocol*: **Email**
   * *Endpoint*: Enter your email address
7. Click **Create subscription**.
8. Check your email.
9. In the subscription confirmation email, click **Confirm subscription**.

**Set Up S3 Bucket as Source Repository**

1. Navigate to S3.
2. Click **Create bucket**.
3. Give it a name (it must be globally unique).
4. Click **Next** through the following screens, leaving the defaults.
5. Click **Create bucket**.
6. Click to open the bucket.
7. Click **Upload**.
8. Upload the MessageUtil.zip file provided with the [lab GitHub page](https://github.com/natonic/Developer-Tools-Deep-Dive/tree/master/Labs/MonitorCodepipelineThruCloudWatch).
9. Accept the defaults, and click **Upload**.

**Create a CodePipeline Pipeline**

1. In a new browser tab, navigate to CodePipeline.
2. Click **Create pipeline**.
3. Give it a name of "pipeline4lab".
4. Leave the other settings as-is, and click **Next**.
5. On the *Add source stage* page, set the following values:
   * *Source provider*: **Amazon S3**
   * *Bucket*: Select your bucket name
   * *S3 object key*: **MessageUtil.zip**
6. Click **Next**.
7. Click **Skip build stage** and then **Skip** in the dialog box.
8. On the *Add deploy stage* page, set the following values:
   * *Deploy provider*: **Amazon S3**
   * *Region*: **US East - (N. Virginia)**
   * *Bucket*: Select your bucket name
   * *S3 object key*: **MessageUtil.zip**
9. Click **Next**.
10. Click **Create pipeline**.
11. In the *Source* failure message area, click **Details**. We should see an error dialog box that says, "The source artifact bucket <YOUR BUCKET NAME> is not versioned."
12. Back in the S3 browser tab, navigate to your bucket's **Properties** tab.
13. Select the **Versioning** card.
14. Select **Enable versioning** and click **Save**.
15. Back in CodePipeline, click **Retry**.
16. Note that the Amazon S3 version id listed changes every few seconds.

**Create a Rule and Add the SNS Topic as the Target**

1. In a new browser tab, navigate to **CloudWatch** > **Events**.
2. Click **Get started**.
3. In the *Event Source* section, select **Event Pattern**.
4. Set *Service Name* to **CodePipeline**.
5. Set *Event Type* to **CodePipeline Pipeline Execution State Change**.
6. Click **Add target**.
7. In the dropdown, select **SNS topic**.
8. Set *Topic* to **pipelineactivity**.
9. Click **Configure details**.
10. Give the rule a *Name* of "pipelineevents".
11. Click **Create rule**.
12. Back in CodePipeline, click **Release change**.
13. Check your email to monitor for notifications.
14. In CodePipeline, click **Disable transition**.
15. In the *Disable transition* dialog box, give a *Disabling reason* of "need to stop loop", and click **Disable**.
16. Notice the S3 version ids are no longer the same. If it were still running, they would keep matching.

**Conclusion**

Congratulations on successfully completing this hands-on lab!