## The Subscription Filters with Kinesis

## The steps outlined in this document would push the cloudwatch logs into the Kinesis stream:

1. **Create a destination Kinesis stream using the following command:**

[root@ip-172-31-94-89 ~]$ aws kinesis create-stream --stream-name "CW-Kinesis-Test" --shard-count 1

You must specify a region. You can also configure your region by running "aws configure".

[root@ip-172-31-94-89 ~]$

**2. If the above command returns “You must specify a region. You can also configure your region by running "aws configure.”, then run the below command and key in your details as below:**

[root@ip-172-31-94-89 ~]$ aws configure

AWS Access Key ID [None]: AKIAJV3B5IJNN6N4VIAA

AWS Secret Access Key [None]: zHrNZOeLu/+Sm1RSlLUUJddXEsttcLdHMsVcZTdI

Default region name [None]: us-east-1

Default output format [None]:

[root@ip-172-31-94-89 ~]$

**3. Ensure that the stream status is ACTIVE. In addition, note the StreamDescription.StreamARN value, as you will need it in a later step:**

[root@ip-172-31-94-89 ~]$ aws kinesis describe-stream --stream-name "CW-Kinesis-Test"

{

"StreamDescription": {

"KeyId": null,

"EncryptionType": "NONE",

"StreamStatus": "ACTIVE",

"StreamName": "CW-Kinesis-Test",

"Shards": [

{

"ShardId": "shardId-000000000000",

"HashKeyRange": {

"EndingHashKey": "340282366920938463463374607431768211455",

"StartingHashKey": "0"

},

"SequenceNumberRange": {

"StartingSequenceNumber": "49596411582086939946562863768253695233011341167775711234"

}

}

],

"StreamARN": "arn:aws:kinesis:us-east-1:931012476720:stream/CW-Kinesis-Test",

"EnhancedMonitoring": [

{

"ShardLevelMetrics": []

}

],

"StreamCreationTimestamp": 1559797303.0,

"RetentionPeriodHours": 24

}

}

**4. Create the IAM role that will grant CloudWatch Logs permission to put data into the Kinesis stream.**

[root@ip-172-31-94-89 ~]$ vi ~/TrustPolicyForCWLtoKIN.json

paste the below content in it and save the file

{

"Statement": {

"Effect": "Allow",

"Principal": { "Service": "logs.us-east-1.amazonaws.com" },

"Action": "sts:AssumeRole"

}

}

**5. Create the role and note the returned Role.Arn value, as you will also need it for a later step:**

[root@ip-172-31-94-89 ~]$ aws iam create-role --role-name CWLtoKinRole --assume-role-policy-document file://~/TrustPolicyForCWLtoKIN.json

{

"Role": {

"AssumeRolePolicyDocument": {

"Statement": {

"Action": "sts:AssumeRole",

"Effect": "Allow",

"Principal": {

"Service": "logs.us-east-1.amazonaws.com"

}

}

},

"RoleId": "AROA5RRFK74YM4YC75DRB",

"CreateDate": "2019-06-06T05:10:58Z",

"RoleName": "CWLtoKinRole",

"Path": "/",

"Arn": "arn:aws:iam::931012476720:role/CWLtoKinRole"

}

}

**6. Create a permissions policy to define what actions CloudWatch Logs can do on your account.**

vi ~/PermissionsForCWLtoKIN.json

paste the below content in it and save the file

{

"Statement": [

{

"Effect": "Allow",

"Action": "kinesis:PutRecord",

"Resource": "arn:aws:kinesis:us-east-1:931012476720:stream/CW-Kinesis-Test"

},

{

"Effect": "Allow",

"Action": "iam:PassRole",

"Resource": "arn:aws:iam::931012476720:role/CWLtoKinRole"

}

]

}

**7. Create a permissions policy to define what actions CloudWatch logs can do on the account.**

[root@ip-172-31-94-89 ~]$ aws iam create-policy --policy-name Permissions-Policy-For-CWLtoKIN --policy-document file://~/PermissionsForCWLtoKIN.json

{

"Version": "2012-10-17",

"Policy": {

"PolicyName": "Permissions-Policy-For-CWLtoKIN",

"PermissionsBoundaryUsageCount": 0,

"CreateDate": "2019-06-06T05:56:07Z",

"AttachmentCount": 0,

"IsAttachable": true,

"PolicyId": "ANPA5RRFK74YPEU73RQFD",

"DefaultVersionId": "v1",

"Path": "/",

"Arn": "arn:aws:iam::931012476720:policy/Permissions-Policy-For-CWLtoKIN",

"UpdateDate": "2019-06-06T05:56:07Z"

}

}

[root@ip-172-31-94-89 ~]$

**8. Associate the permissions policy with the role**

[root@ip-172-31-94-89 ~]$ aws iam put-role-policy --role-name CWLtoKinRole --policy-name Permissions-Policy-For-CWLtoKIN --policy-document file://~/PermissionsForCWLtoKIN.json

**9. After the Kinesis stream is in Active state and you have created the IAM role, you can create the CloudWatch Logs subscription filter. The subscription filter immediately starts the flow of real-time log data from the chosen log group to your Kinesis stream:**

aws logs put-subscription-filter --log-group-name "/aws/lambda/LogsToElasticsearch\_elasticsearch-log" --filter-name "CW-Kinesis-Test" --filter-pattern "{$.userIdentity.type = Root}" --destination-arn "arn:aws:kinesis:us-east-1:931012476720:stream/CW-Kinesis-Test" --role-arn "arn:aws:iam::931012476720:role/CWLtoKinRole"

**10. After you set up the subscription filter, CloudWatch Logs forwards all the incoming log events that match the filter pattern to your Kinesis stream. You can verify that this is happening by grabbing a Kinesis shard iterator and using the Kinesis get-records command to fetch some Kinesis records:**

aws kinesis get-shard-iterator --stream-name CW-Kinesis-Test --shard-id shardId-000000000000 --shard-iterator-type TRIM\_HORIZON

[root@ip-172-31-94-89 ~]# aws kinesis get-shard-iterator --stream-name CW-Kinesis-Test --shard-id shardId-000000000000 --shard-iterator-type TRIM\_HORIZON

{

"ShardIterator": "AAAAAAAAAAGYL60VeV0bf9634doVd9KG4yxjrrFXn24Up3BWFwqSB+DtWLP8fo59Hlh1PthGezZv8ZxVYjNiT65nw3iH7plVIfejWfa3aB6lTlrY83uh0X7oYSFi4/G22TNtTctuFDcjjmpwVNPAFtAhU9aXG0B5GaGZoDTKYjdKA1voxAUsMdeS2qTcAIuXLg4BE2GpU0mS0t9fKztnYr7nweSjOsdx"

}

[root@ip-172-31-94-89 ~]#

[root@ip-172-31-94-89 ~]# aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAGYL60VeV0bf9634doVd9KG4yxjrrFXn24Up3BWFwqSB+DtWLP8fo59Hlh1PthGezZv8ZxVYjNiT65nw3iH7plVIfejWfa3aB6lTlrY83uh0X7oYSFi4/G22TNtTctuFDcjjmpwVNPAFtAhU9aXG0B5GaGZoDTKYjdKA1voxAUsMdeS2qTcAIuXLg4BE2GpU0mS0t9fKztnYr7nweSjOsdx"

{

"Records": [

{

"Data": "H4sIAAAAAAAAADWOwQqCQBRFf2WYdURFkboLsRZZQgYtQmLSlz7SGZk3JhH+e6PW8nAv954Pr4BI5HB+18A97kfH8ykKb4cgjje7gE+4aiXoPilVk7XCpEWocrJBqfKdVk1ts5Fio0FUI1Jzp1RjbVDJLZYGNHHvmgy94AXS9PjhmI11g1bDiMqOzVcr15m5rrNcrJ3JX68XuITsp8d+eh7zC0ifKHNWgChNwdSDZXYJpeif2R4lEBKjQW3Ku6T7AhyrdBzyAAAA",

"PartitionKey": "3e21f5e8240cbb048271af4fdb892a1c",

"ApproximateArrivalTimestamp": 1559809984.34,

"SequenceNumber": "49596415852768848445971317873220043402294995525102993410"

},

{

"Data": "H4sIAAAAAAAAADWOXQuCQBBF/8qyzxJ9CeVbiPWQJWTQQ0hsOumQ7srOmoT031u1XgYO93LndLwCIpHD+V0D97gfHc+nKLwdgjje7ALucNVK0H1SqiZrhUmLUOVkg1LlO62a2mYjxUaDqEak5k6pxtqgklssDWji3jUZesELpOmx45iNdYNWw4jKjs1cd72aTRf2LufOX68XuITsp8d+eh7zC0ifKHNWgChNwdSDZXYJpeg/sz1KICRGg9qEf5LPFwUxB/jyAAAA",

"PartitionKey": "3e21f5e8240cbb048271af4fdb892a1c",

"ApproximateArrivalTimestamp": 1559810359.903,

"SequenceNumber": "49596415852768848445971317873486007082610239713342128130"

},

{

"Data": "H4sIAAAAAAAAADWOwQqCQBRFf2WYdURGQrqLMBdZQgotImLS1/hIZ2TemET0741Zy8O93HtevAEiISF/tsBDvk73+SFNLrsoy1ZxxCdc9wrMkNS6K3thiyrRklxQaxkb3bUuGymzBkQzInVXKgy2FrXaYG3BEA9P528veoCyA744lmPdotOwonFjnu8HS2+2mPvzwJv89QaBY8J+euynF7J1BcUdlWQViNpWTN9Y6ZZQieGZbVEBITH6qk35+/z+ABmNI6HyAAAA",

"PartitionKey": "3e21f5e8240cbb048271af4fdb892a1c",

"ApproximateArrivalTimestamp": 1559810425.344,

"SequenceNumber": "49596415852768848445971317873487216008429858878002298882"

}

],

"NextShardIterator": "AAAAAAAAAAGKM74Z7LLmFXafRZ2LxCWwIEOqfUW/AaesE4N67ZVG3h4M9gGLBXfx9bUIaMYs3M0I2SaaU2K35iGg1gr2W1nOEDNHLIulkKYFZVa5ob5sXrZyzw8O/vPAUqykqK7ZlMiWPzRs+SGbOF4bPCT1P0nnNPJ/RxFjjBfgokgzVG4UMhwNFjmjbXq176weEin0pJMnTftiv7RcLa4VkXvKY47z",

"MillisBehindLatest": 0

}

[root@ip-172-31-94-89 ~]#

**11. Note that you might need to make this call a few times before Kinesis starts to return data.**

**You should expect to see a response with an array of records. The Data attribute in a Kinesis record is Base64 encoded and compressed with the gzip format. You can examine the raw data from the command line using the following Unix commands:**

[root@ip-172-31-94-89 ~]# echo -n "H4sIAAAAAAAAADWOXQuCQBBF/8qyzxJ9CeVbiPWQJWTQQ0hsOumQ7srOmoT031u1XgYO93LndLwCIpHD+V0D97gfHc+nKLwdgjje7ALucNVK0H1SqiZrhUmLUOVkg1LlO62a2mYjxUaDqEak5k6pxtqgklssDWji3jUZesELpOmx45iNdYNWw4jKjs1cd72aTRf2LufOX68XuITsp8d+eh7zC0ifKHNWgChNwdSDZXYJpeg/sz1KICRGg9qEf5LPFwUxB/jyAAAA" | base64 -d | zcat

{"messageType":"CONTROL\_MESSAGE","owner":"CloudwatchLogs","logGroup":"","logStream":"","subscriptionFilters":[],"logEvents":[{"id":"","timestamp":1559810359842,"message":"CWL CONTROL MESSAGE: Checking health of destination Kinesis stream."}]}[root@ip-172-31-94-89 ~]#