ECV-BD-312

How to Visualize and Refine Your Network's Security by Adding Security Group IDs to Your VPC Flow Logs

2018.2.2

Version 1.0

Agenda

About this lab 3

Scenario 3

Prerequisites 4

Lab tutorial 4

Create your Amazon ES cluster and VPC Flow Logs 4

Enable VPC flow logs 5

Set up AWS Lambda to enrich the VPC Flow Logs dataset with security group IDs 6

Connect to your Linux instance (From Windows client) 7

Install and Setup into Amazon Linux instance 9

Set up Firehose 10

Stream data to Firehose 11

Using the SGDashboard to analyze VPC network traffic 12

Conclusion 14

About this lab

Scenario

Many organizations begin their cloud journey to AWS by moving a few applications to demonstrate the power and flexibility of AWS. This initial application architecture includes building security groups that control the network ports, protocols, and IP addresses that govern access and traffic to their AWS Virtual Private Cloud (VPC). When the architecture process is complete and an application is fully functional, some organizations forget to revisit their security groups to optimize rules and help ensure the appropriate level of governance and compliance. Not optimizing security groups can create less-than-optimal security, with ports open that may not be needed or source IP ranges set that are broader than required.

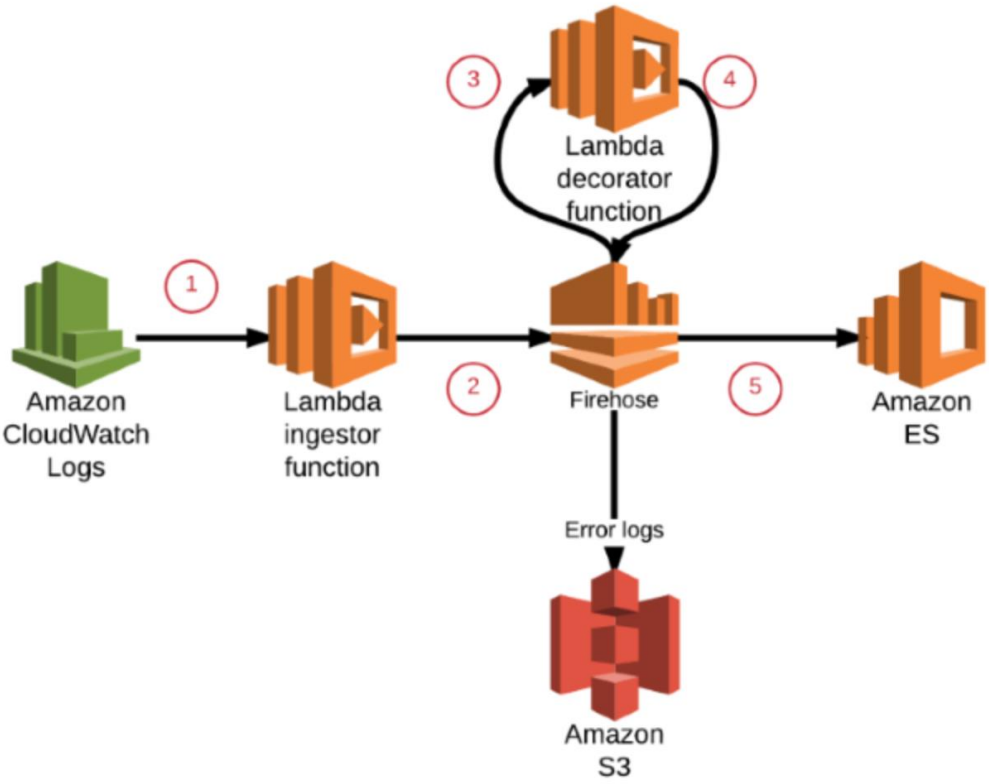


Figure 1 Architecture

As illustrated in the preceding diagram, this is how the data flows in this model:

1. The VPC posts its flow log data to Amazon CloudWatch Logs.
2. The Lambda ingestor function passes the data to Firehose.
3. Firehose then passes the data to the Lambda decorator function.
4. The Lambda decorator function performs a number of lookups for each record and returns the data to Firehose with additional fields.
5. Firehose then posts the enhanced dataset to the Amazon ES endpoint and any errors to Amazon S3.

The workshop's region will be in 'Oregon'

Prerequisites

- Download Putty: IF you don't already have the **PuTTy client**/**PuTTYgen** installed on your machine, you can download and then launch it from here: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Lab tutorial

Create your Amazon ES cluster and VPC Flow Logs

1.1. In the **AWS Management Console**, on the **service** menu, choose **Elasticsearch** Service under Analytics.

1.2. Choose Create a new domain or Get started.

1.3. Type es-flowlogs for the **Elasticsearch domain name.**

1.4. Set **Version** to **5.1** in the drop-down list. Choose **Next**.

1.5. Set **Instance count** to **1** and set **Instance type** to **t2.small.elasticsearch.** Choose **Next.**

1.6. For Network configuration, select **Public Access.**

1.7. For Access Policy, Set the domain access policy to **Allow open access to the domain.** Click **I accept the risk** and choose **OK.**

1.8. Choose **Next.**

1.9. On the next page, choose **Confirm**.

Enable VPC flow logs

1.10. In the AWS Management Console, choose **CloudWatch** under **Management Tools**.

1.11. Click **Logs** in the navigation pane.

1.12. From the Actions drop-down list, choose **Create log group.**

1.13. Type **Flowlogs** as the **Log Group Name**.

1.14. In the AWS Management Console, choose **VPC** under **Networking & Content Delivery.**

1.15. Choose **Your VPCs** in the navigation pane, and select the VPC you would like to analyze.

1.16. Choose the **Flow Logs** tab in the bottom pane, and then choose **Create Flow Log.**

1.17. In the text beneath the **Role** box, choose **Set Up Permissions** (this will open an IAM management page).

1.18. Choose **Allow** on the **IAM management** page. Return to the VPC Flow Logs setup page.

1.19. Choose **All** from the **Filter** drop-down list.

1.20. Choose **flowlogsRole** from the **Role** drop-down list (you created this role in steps 3 and 4 in this procedure).

1.21. Choose **Flowlogs** from the **Destination Log Group** drop-down list.

1.22. Choose **Create Flow Log**.

Set up AWS Lambda to enrich the VPC Flow Logs dataset with security group IDs

1.23. Update a In the **AWS Management Console**, on the **service** menu, click **EC2**.

1.24. Click **Launch Instance**.

1.25. In the navigation pane, choose **Quick Start**, in the row for **Amazon Linux AMI**, click **Select**.

1.26. On **Step2: Choose a Instance Type** page, make sure **t2.micro** is selected and click **Next: Configure Instance Details**.

1.27. On **Step3: Configure Instance Details** page, enter the following and leave all other values with their default:

1.28. Network: Default VPC

1.29. Subnet: No preference

1.30. Auto-assign Public IP: click **Enable**

1.31. Click **Next: Add Storage**, leave all values with their default.

1.32. Click **Next: Tag Instance**.

1.33. On **Step5: Tag Instance** page, enter the following information:

Key: Name

Value: Lab Server

1.34. Click **Next: Configure Security Group**.

1.35. On **Setp6: Configure Security Group** page, click **create a new security group**, enter the following information:

1.36. Security group name: LabSecurityGroup

1.37. **Description: Enable SSH, HTTP and HTTPS access**

1.38. Click **Add Rule**.

1.39. For Type, click **SSH (22), HTTP (80)**.

1.40. Click **Review and Launch**.

1.41. Review the instance information and click Launch.

1.42. Click **Create a new key pair**, enter the **Key pair name (ex. amazonec2_keypair_oregon)**, click **Download Key Pair**.

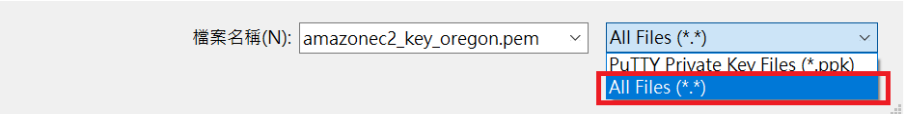
1.43. Click **Launch Instances**.

1.44. Scroll down and click **View Instances**.

1.45. Wait until **Lab Server** shows 2/2 checks passed in the **Status Checks** column. This will take 3-5 minutes. Use the refresh icon at the top right to check for updates.

Connect to your Linux instance (From Windows client)

1.46. Start PuTTYgen.exe, click **Load**. By default, PuTTYgen display only files with the extension *.ppk*. to locate your *.pem* file, select the option to display files of all types.



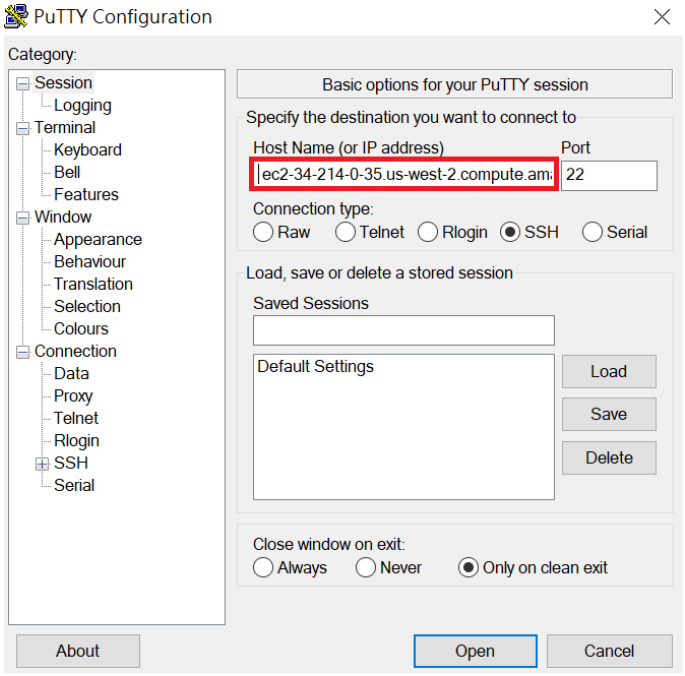
1.47. Select your *.pem* file (**ex. amazonec2_keypair_oregon.pem**), and then click **Open**. Click **OK** to dismiss the confirmation dialog box.

1.48. Click **Save private key** to save the key in the format that PuTTY can use. PuTTYgen displays a warning about saving the key without a passphrase, click **Yes**.

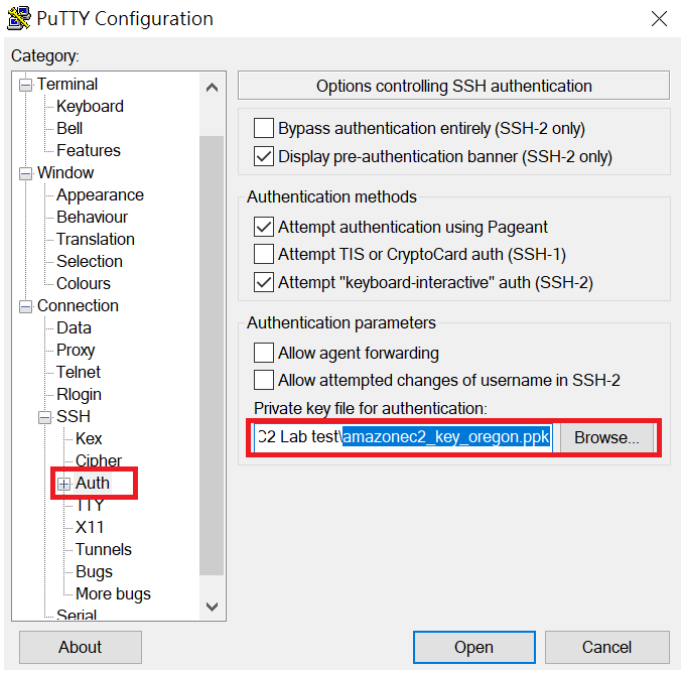
1.49. Specify the same name for the key that you used for the key pair (ex. **amazonec2_keypair_oregon.ppk**). PuTTY automatically adds the *.ppk* extension.

1.50. Start **PuTTY.exe**, enter **Host Name**, Select Lab Server, and copy the **public IP**

value.

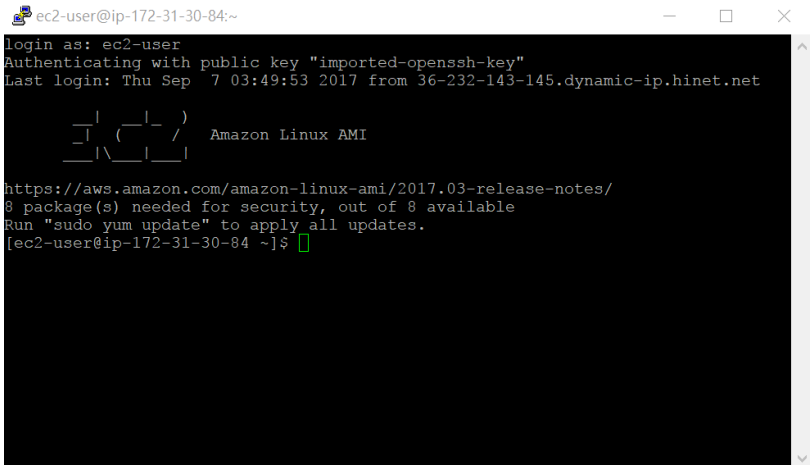


1.51. On the navigation pane, click **Connect>SSH>Auth**, click **Browse** to choose your key pair (ex. **amazonec2_keypair_oregon.ppk**), click **Open**.



1.52. Enter **ec2-user**,

1.53. You are successfully connecting to EC2.



Install and Setup into Amazon Linux instance

1.54. Install NPM into Amazon Linux instance

```
[ec2-user ~]$ wget
-qO- https://raw.githubusercontent.com/creationix/nvm/
v0.33.8/install.sh | bash
[ec2-user ~]$ . ~/.nvm/nvm.sh
[ec2-user ~]$ nvm install 6.11.5
[ec2-user ~]$ command -v nvm
                    //you will see nvm as output, then it should install success
```

1.55. Install GIT and Prepare to deploy lambda

```
[ec2-user ~]$ sudo yum install git
[ec2-user ~]$ git
clone https://github.com/awslabs/aws-vpc-flow-log-appender
[ec2-user ~]$ cd aws-vpc-flow-log-appender/decorator
[ec2-user ~]$ npm install
[ec2-user ~]$ cd ../ingestor
[ec2-user ~]$ npm install
[ec2-user ~]$ cd ..
[ec2-user ~]$ aws configure
//Enter Information as below
ACCESS_KEY
Secret_ACCESS_Key
Region : us-west-2
Format : json
```

1.56. Deploy Lambda Functions and Create buckets

```
[ec2-user ~]$ aws s3 mb s3://YOUR_BUCKET_NAME
[ec2-user ~]$ aws cloudformation package --template-file
app-sam.yaml --s3-bucket YOUR_BUCKET_NAME
--output-template-file app-sam-output.yaml
[ec2-user ~]$ aws cloudformation deploy --template-file
app-sam-output.yaml --stack-name
vpc-flow-log-appender-dev --capabilities CAPABILITY_IAM
```

Set up Firehose

1.57. In the AWS Management Console, choose **Kinesis** under Analytics.

1.58. Choose Go to **Firehose** and then choose **Create Delivery Stream.**

1.59. For Delivery stream name, type **VPCFlowLogsToElasticSearch** (the name must match the default environment variable in the ingestion Lambda function). Choose **Next**.

1.60. For **Transform source records**, choose **Enabled**.

1.61. Choose **vpc-flow-log-appender-dev-FlowLogDecoratorFunction-xxxxx** from the **Lambda function** drop-down list (make sure you select the Decorator function). Choose Next.

1.62. Choose **Amazon Elasticsearch Service** from Destination.

1.63. Choose **es-flowlogs** from the Elasticsearch domain drop-down list. (The Amazon ES cluster configuration state must be Active for es-flowlogs to be available in the drop-down list.)

1.64. For **Index**, type **cwl.**

1.65. Choose **Every day** from the Index rotation drop-down list.

1.66. For **Type**, type **log**.

1.67. For S3 Backup, choose **Failed Documents Only**.

1.68. For Backup S3 bucket, choose S3 bucket name from the drop-down list, or choose Create S3 bucket. Choose **Next**.

1.69. Under IAM role, choose **Create new, or Choose.**

1.70. Choose **Allow**. This takes you back to the Firehose Configuration.

1.71. Choose **Next**, and then choose **Create Delivery Stream**.

Stream data to Firehose

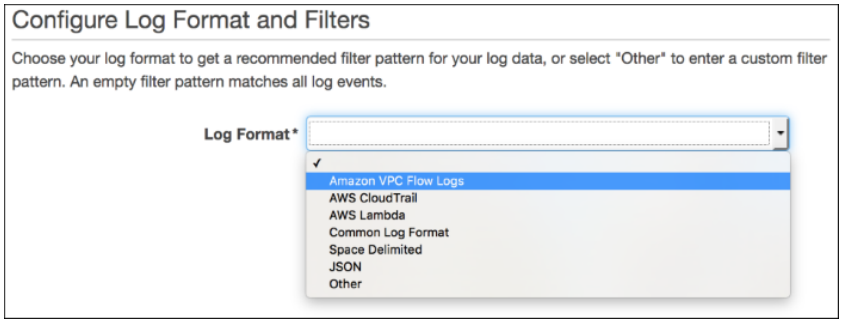
1.72. In the AWS Management Console, choose **CloudWatch** under Management Tools.

1.73. Choose Logs in the navigation pane, and select the check box next to **Flowlogs** under Log Groups.

1.74. From the Actions menu, choose **Stream to AWS Lambda.** Choose **vpc-flow-log-appender-dev-FlowLogIngestionFunction-xxxxx**

xx (select the Ingestion function). Choose Next.

1.75. Choose **Amazon VPC Flow Logs** from the Log Format drop-down list.



1.76. Choose **Next**.

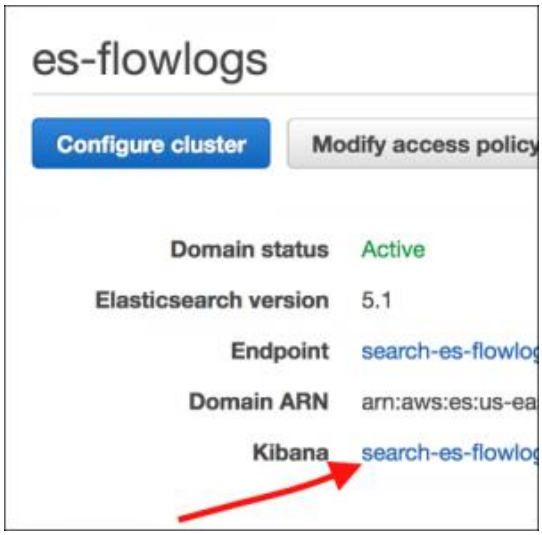
1.77. Choose **Start Streaming**.

Using the SGDashboard to analyze VPC network traffic

1.78. In the AWS Management Console, click **Elasticsearch** Service under Analytics.

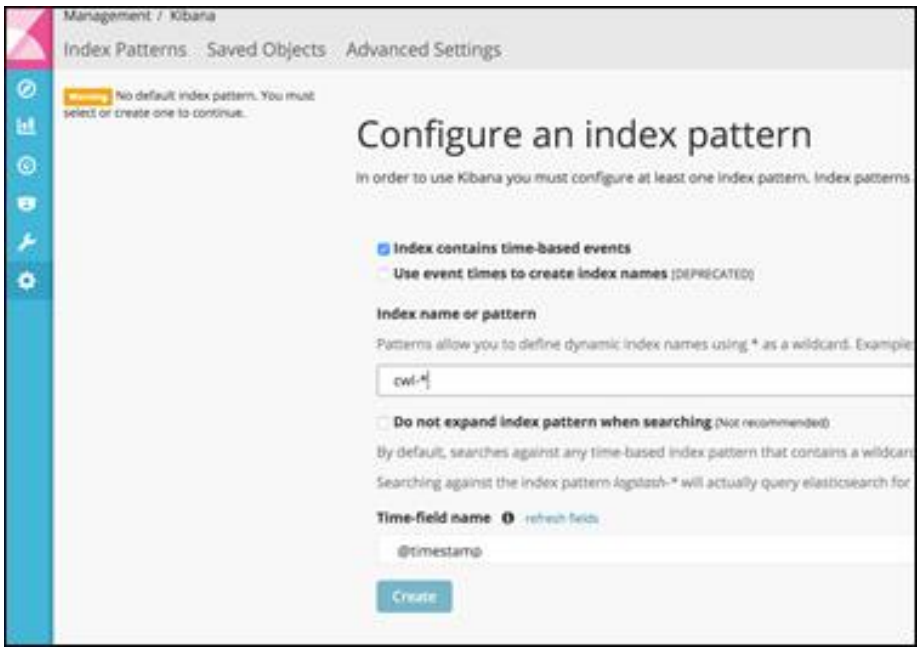
1.79. Choose es-**flowlogs** under Elasticsearch domain name.

1.80. Click the link next to **Kibana**, as shown in the following screenshot.



The first time you access Kibana, you will be asked to set the defaultindex. To set the defaultindex in the Amazon ES cluster:

1.81. Set the Index name or pattern to **cwl-*.**



1.82. For Time-field name, type **@timestamp.**

1.83. Choose **Create**.

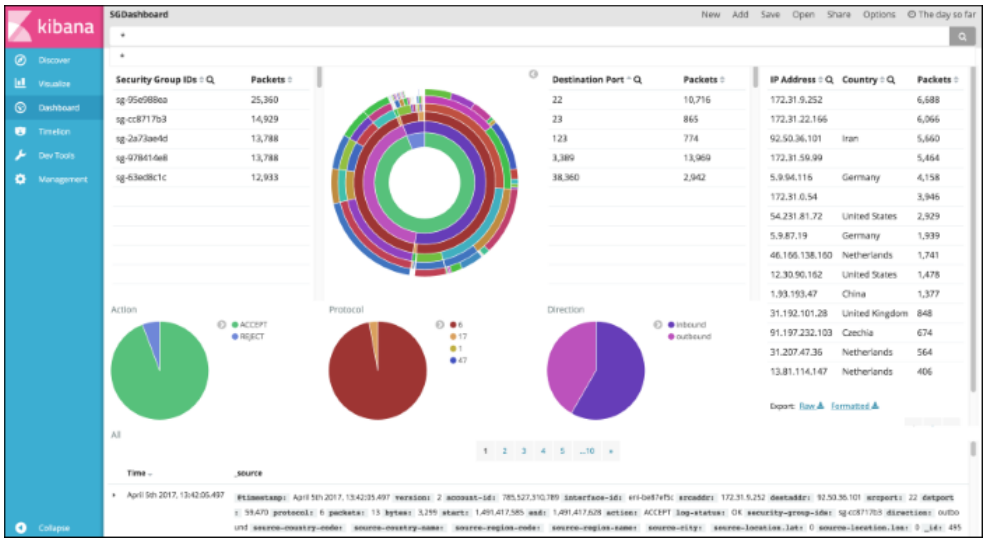
Load the SGDashboard:

1.84. Download this JSON file and save it to your computer. The file includes a dashboard and visualizations. Download Link: https://s3-us-west-2.amazonaws.com/aws-ecv-training/FlowLogDashboard.json

1.85. In Kibana, choose **Management** in the navigation pane, choose **Saved Objects**, and then import the file you just downloaded.

1.86. Choose **Dashboard** and Open to **load the SGDashboard** you just imported. (You might have to press Enter in the top search box to have the dashboard load the first time.)

1.87. The following screenshot shows the SGDashboard after it has loaded.



Conclusion

Congratulations! You now have learned how to:

- The VPC posts its flow log data to Amazon CloudWatch Logs.
- The Lambda ingestor function passes the data to Firehose.
- Firehose then passes the data to the Lambda decorator function.
- The Lambda decorator function performs a number of lookups for each record and returns the data to Firehose with additional fields.
- Firehose then posts the enhanced dataset to the Amazon ES endpoint and any errors to Amazon S3.