

Logstash Azure Event Hub

 chatgpt.com/c/57ecbfbe-683f-4d96-9511-04319401d12b

Send logs from Logstash to the Azure Event Hub and save those logs in ADLS parquet format, Please give me end to end details, i'm new to this project.

Step 1: Set Up Azure Event Hub

1. Create an Event Hub Namespace:

- Go to the Azure portal.
- Search for "Event Hubs" and create a new Event Hub namespace.
- Provide the required details (e.g., name, region, pricing tier) and create it.

2. Create an Event Hub:

- Within the namespace, create an Event Hub.
- Configure the necessary settings (e.g., partition count, retention period).

Step 2: Configure Logstash

1. Install Logstash:

Follow the instructions to [install Logstash](#) on your machine.

2. Install Azure Event Hubs Plugin:

Run the following command to install the Azure Event Hubs output plugin for Logstash:

```
sh
```

```
bin/logstash-plugin install logstash-output-azure_event_hubs
```

3. Create Logstash Configuration File:

- Create a configuration file (e.g., `logstash.conf`) with the following content:

plaintext

```
input {
  file {
    path => "/path/to/your/logs/*"
    start_position => "beginning"
  }
}

output {
  azure_event_hubs {
    event_hub_name => "your-event-hub-name"
    namespace => "your-namespace"
    sas_policy => "your-sas-policy-name"
    sas_key => "your-sas-key"
  }
}
```

- Replace `your-event-hub-name`, `your-namespace`, `your-sas-policy-name`, and `your-sas-key` with the actual values from your Azure Event Hub.

Step 3: Send Logs to Azure Event Hub

1. Run Logstash:

Start Logstash with your configuration file:

sh

```
bin/logstash -f logstash.conf
```

Step 4: Set Up Azure Data Lake Storage (ADLS)

1. Create an ADLS Gen2 Account:

- Go to the Azure portal.
- Search for "Storage accounts" and create a new storage account.
- Ensure you enable the "Hierarchical namespace" feature for ADLS Gen2.

2. Create a Container:

Inside your storage account, create a container to store your logs.

Step 5: Process and Save Logs in Parquet Format

1. Set Up Azure Databricks:

- Create an Azure Databricks workspace.
- Launch a Databricks cluster.

2. Configure Event Hub Integration:

Use Databricks to read from Azure Event Hub:

```
python
```

```
from pyspark.sql import SparkSession

spark =
SparkSession.builder.appName("EventHubIntegration").getOrCreate()

connectionString = "your-event-hub-connection-string"
eventHubsConf = {
    'eventhubs.connectionString': connectionString
}

df =
spark.readStream.format("eventhubs").options(**eventHubsConf)
.load()
```

3. Transform and Write to ADLS in Parquet Format:

Transform the data and write it to ADLS in Parquet format:

```
python

outputPath = "abfss://your-container@your-storage-
account.dfs.core.windows.net/logs-parquet"

query = df.writeStream \
    .format("parquet") \
    .option("path", outputPath) \
    .option("checkpointLocation", "/path/to/checkpoint/dir") \
    .start()

query.awaitTermination()
```

Summary

1. Set up Azure Event Hub and ADLS.
2. Configure Logstash to send logs to Event Hub.
3. Use Azure Databricks to read from Event Hub and write logs to ADLS in Parquet format.

Feel free to ask for further details on any of these steps!

I was collecting logs from on-premises window servers.

To collect logs from on-premises Windows servers and send them to Azure Event Hub using Logstash, you need to configure Logstash to read from the Windows Event Log. Here's an end-to-end guide:

Step 1: Set Up Azure Event Hub

1. Create an Event Hub Namespace:

- Go to the Azure portal.
- Search for "Event Hubs" and create a new Event Hub namespace.
- Provide the required details (e.g., name, region, pricing tier) and create it.

2. Create an Event Hub:

- Within the namespace, create an Event Hub.
- Configure the necessary settings (e.g., partition count, retention period).

Step 2: Configure Logstash on Windows Server

1. Install Logstash:

Download and install Logstash on your Windows server from the [official Logstash download page](#).

2. Install Azure Event Hubs Plugin:

Open Command Prompt and run the following command to install the Azure Event Hubs output plugin for Logstash:

```
sh
```

```
bin\logstash-plugin install logstash-output-azure_event_hubs
```

3. Create Logstash Configuration File:

- Create a configuration file (e.g., `logstash.conf`) with the following content:

```
plaintext
```

```
input {
  winlogbeat {
    type => 'wineventlog'
    event_logs => ['Application', 'System']
  }
}

output {
  azure_event_hubs {
    event_hub_name => "your-event-hub-name"
    namespace => "your-namespace"
    sas_policy => "your-sas-policy-name"
    sas_key => "your-sas-key"
  }
}
```

- Replace `your-event-hub-name`, `your-namespace`, `your-sas-policy-name`, and `your-sas-key` with the actual values from your Azure Event Hub.

Step 3: Send Logs to Azure Event Hub

1. Run Logstash:

Start Logstash with your configuration file:

```
sh
```

```
bin\logstash -f logstash.conf
```

Step 4: Set Up Azure Data Lake Storage (ADLS)

1. Create an ADLS Gen2 Account:

- Go to the Azure portal.
- Search for "Storage accounts" and create a new storage account.
- Ensure you enable the "Hierarchical namespace" feature for ADLS Gen2.

2. Create a Container:

Inside your storage account, create a container to store your logs.

Step 5: Process and Save Logs in Parquet Format

1. Set Up Azure Databricks:

- Create an Azure Databricks workspace.
- Launch a Databricks cluster.

2. Configure Event Hub Integration:

Use Databricks to read from Azure Event Hub:

```
python
```

```
from pyspark.sql import SparkSession

spark =
SparkSession.builder.appName("EventHubIntegration").getOrCreate()

connectionString = "your-event-hub-connection-string"
eventHubsConf = {
    'eventhubs.connectionString': connectionString
}

df =
spark.readStream.format("eventhubs").options(**eventHubsConf)
.load()
```

3. Transform and Write to ADLS in Parquet Format:

Transform the data and write it to ADLS in Parquet format:

```
python
```

```
outputPath = "abfss://your-container@your-storage-  
account.dfs.core.windows.net/logs-parquet"  
  
query = df.writeStream \  
    .format("parquet") \  
    .option("path", outputPath) \  
    .option("checkpointLocation", "/path/to/checkpoint/dir") \  
    .start()  
  
query.awaitTermination()
```

Summary

1. Set up Azure Event Hub and ADLS.
2. Configure Logstash to collect logs from Windows Event Log and send them to Event Hub.
3. Use Azure Databricks to read from Event Hub and write logs to ADLS in Parquet format.

If you need further details on any of these steps, feel free to ask!

Logstash configuration file I need some better understanding about input, defining single server, multiple server etc.

Sure, I'll provide more details on configuring Logstash to collect logs from single and multiple Windows servers.

Single Server Configuration

For a single server, you will configure the Logstash input plugin to read from the Windows Event Log. Here's a sample configuration:

1. Logstash Configuration File for Single Server:

plaintext

```
input {
  # Read from Windows Event Log
  winlogbeat {
    type => 'wineventlog'
    event_logs => ['Application', 'System', 'Security'] # Specify
the logs you want to collect
  }
}

output {
  azure_event_hubs {
    event_hub_name => "your-event-hub-name"
    namespace => "your-namespace"
    sas_policy => "your-sas-policy-name"
    sas_key => "your-sas-key"
  }
}
```

Multiple Servers Configuration

For multiple servers, there are several approaches you can take:

1. Centralized Logstash Instance:

- Install Logstash on one central server.
- Configure each Windows server to forward logs to the central Logstash instance.

2. Winlogbeat on Each Server:

- Install Winlogbeat on each Windows server.
- Configure Winlogbeat to send logs to a centralized Logstash instance or directly to Azure Event Hub.

Centralized Logstash with Forwarding

1. Install Winlogbeat on Each Server:

- Download and install Winlogbeat on each Windows server.
- Configure `winlogbeat.yml` on each server to forward logs to Logstash:

yaml

```
winlogbeat.event_logs:
  - name: Application
  - name: System
  - name: Security

output.logstash:
  hosts: ["central-logstash-server:5044"]
```

2. Logstash Configuration on Central Server:

On the central Logstash server, configure the input to receive logs from Winlogbeat and forward them to Azure Event Hub:

plaintext

```
input {
  beats {
    port => 5044
  }
}

output {
  azure_event_hubs {
    event_hub_name => "your-event-hub-name"
    namespace => "your-namespace"
    sas_policy => "your-sas-policy-name"
    sas_key => "your-sas-key"
  }
}
```

Direct Winlogbeat to Azure Event Hub

Alternatively, you can configure Winlogbeat to send logs directly to Azure Event Hub, bypassing Logstash:

1. Install Winlogbeat on Each Server:

- Download and install Winlogbeat on each Windows server.
- Configure `winlogbeat.yml` on each server to send logs to Azure Event Hub:

yaml

```
winlogbeat.event_logs:
  - name: Application
  - name: System
  - name: Security

output.azure_event_hubs:
  event_hub_name: "your-event-hub-name"
  namespace: "your-namespace"
  sas_policy: "your-sas-policy-name"
  sas_key: "your-sas-key"
```

Summary

- **Single Server:** Configure Logstash to read from Windows Event Log directly.
- **Multiple Servers (Centralized Logstash):** Install Winlogbeat on each server to forward logs to a central Logstash instance.
- **Multiple Servers (Direct to Event Hub):** Install Winlogbeat on each server to send logs directly to Azure Event Hub.

Choose the approach that best fits your infrastructure and scalability needs. If you have further questions or need more details on any step, feel free to ask!