eStreamer eNcore for Microsoft Sentinel 3.6.8

First Published: June 1, 2017
Last Updated: Oct 29 2020

Cisco Systems, Inc.   www.cisco.com

Table of Contents

Table of Contents 2
About This eStreamer eNcore Operations Guide v3.6.8........ 4
Revision History 4
Conventions 4
1 Introduction........ 6
1.1 Document Purpose 6
1.2 Background 6
1.3 Application Summary........ 6
1.3.1 eStreamer-eNcore CLI........ 6
1.3.2 Cisco eStreamer eNcore for Splunk (TA-eStreamer) ....**Error! Bookmark not defined.**
1.3.3 Cisco eStreamer eNcore Dashboard for Splunk (eStreamer Dashboard) ...... **Error! Bookmark not defined.**
2 eNcore CLI Prerequisites........ 6
2.1 Python 2.7 Installation........ 7
2.2 pyOpenSSL........ 7
2.3 EPEL Repo Dependency for RHEL 8
24 Running eNcore CLI on Windows 14
3 Installing eStreamer eNcore CLI 14
3.1 Download eStreamer-eNcore-cli-X.YY.tar.gz 14
3.2 Extract Files........**Error! Bookmark not defined.**
3.3 Create (or copy existing) PKCS12 file 15
3.4 Install the PKCS12 File 15
3.6.8 Test 15
4. Running eNcore CLI........ 17
5. Configuration Options........ 19
5.1 Essential Configuration 19
5.2 Advanced Configuration Options........ 20
5.3 Execution........ 23
5.4 Logging 24
6 Troubleshooting and questions 27
6.1 Error messages 31
6.2 Frequently Asked Questions 32

7 Cisco Support 32
8 Appendix A: 33
8.1 FMC eStreamer Certificate Creation 33
8.2 Example Configuration File 35
Trademarks and Disclaimers 37

About This eStreamer eNcore Operations Guide v3.6.8

Author	Sam Strachan (sastrach)
Change Authority	Cisco Systems Advanced Services, Security & Collaboration IDT, Implementation Americas
Content ID	585637
Project ID	852716

Revision History

Revision	Date	Name or User ID	Comments
1.0	06/01/2017	Michelle Jenkins	Initial Release
3.0	08/25/2017	Sam Strachan	Updated for v3.0
3.5	08/13/2018	Richard Clendenning	Updated for v3.5
3.6.8	08/24/2020	Seyed Khadem	Updated for v3.6.8

Conventions

This document uses the following conventions.

Convention	Indication
bold font	Commands and keywords and user-entered text appear in bold font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
String	A non-quoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier font`	Terminal sessions and information the system displays appear in `courier` font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.

Convention	Indication
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note: Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution: Means *reader be careful.* In this situation, you might perform an action that could result in equipment damage or loss of data.

Warning: IMPORTANT SAFETY INSTRUCTIONS

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Regulatory: Provided for additional information and to comply with regulatory and customer requirements.

1 Introduction

1.1 Document Purpose

This document seeks to outline the background and usage of the eStreamer eNcore client in order to assist users with installation and execution.

1.2 Background

The Cisco Event Streamer (i.e. eStreamer) allows users to stream system intrusion, discovery, and connection data from Firepower Management Center or managed device (i.e., the eStreamer server) to external client applications. eStreamer responds to client requests with terse, compact, binary encoded messages that facilitate high performance.

Historically, the eStreamer SDK has been wrapped with some additional code to create separate Perl applications (e.g., the Cisco eStreamer for Splunk app and the CEF agent).

eStreamer eNcore is a multi-platform, multi-process Python application that is compatible with FMC versions 6.0 and above.

1.3 Application Summary

eNcore is an all-purpose client, which requests all possible events from eStreamer, parses the binary content, and outputs events in various formats to support other SIEMs. eNcore was built from scratch in Python with a scalable and fast multi-process architecture. It supports version 6.0 of Firepower Management Center. It was built and tested on CentOS 7, but should work with any Linux distribution that supports the pre-requisites. The software will run on Windows, although, it has not been made production-ready yet.

There are three packages associated with eStreamer eNcore.

1.3.1 eStreamer-eNcore CLI for Sentinel

This is a command line interface for eStreamer eNcore. It runs standalone to request data from the FMC eStreamer server and output its data. The output data format can be:

- key-value pairs designed to maintain compatibility with previous Splunk collectors
- JSON
- CEF which maintains backwards compatibility with the previous cef-agent.

The output can be streamed to files, a TCP or UDP network port, stdout.

2 eNcore CLI Prerequisites

The CLI version of eNcore can be run on either Python 2.7 or Python 3.6+. You must also have a means of splitting the FMC’s PKCS12 file. The default approach is to install pyOpenSSL and let eNcore do the work for you.

Note: The encore.sh script should guide you through all these points if you wish to get going immediately, but it is worth being familiar with these points prior to install.

To check whether Python2.7 is present, use following command:

```
which python
```

To test where Python2.7 is present, use the following command.

```
whereis python
```

Note: If you are installing the CLI version on a device running Splunk, then it is worth noting that Splunk has its own version of Python. The Splunk Python has been compiled differently from the normal distribution – specifically, it is built with PyUnicodeUCS2. The encore.sh script will detect this and warn you. If you encounter this problem, then you will need to create a new user and run eStreamer-eNcore as that user. You should consider running the Splunk add on instead.

To check for pyOpenSSL, use the following command:

```
pip list | grep -i pyOpenSSL
```

Alternatively using the python3 version will no longer require the pyUnicodeUS4 complication. To access the python3 branch perform the following

```
git checkout python3
```

2.1 Python 2.7 Installation

Use the following command to install Python on CentOS:

```
sudo yum install python
```

2.2 pyOpenSSL

Install pyOpenSSL as follows:

```
sudo yum install python-pip python-devel openssl-devel gcc
sudo pip install pyOpenSSL
```

If using python3 branch then run the following

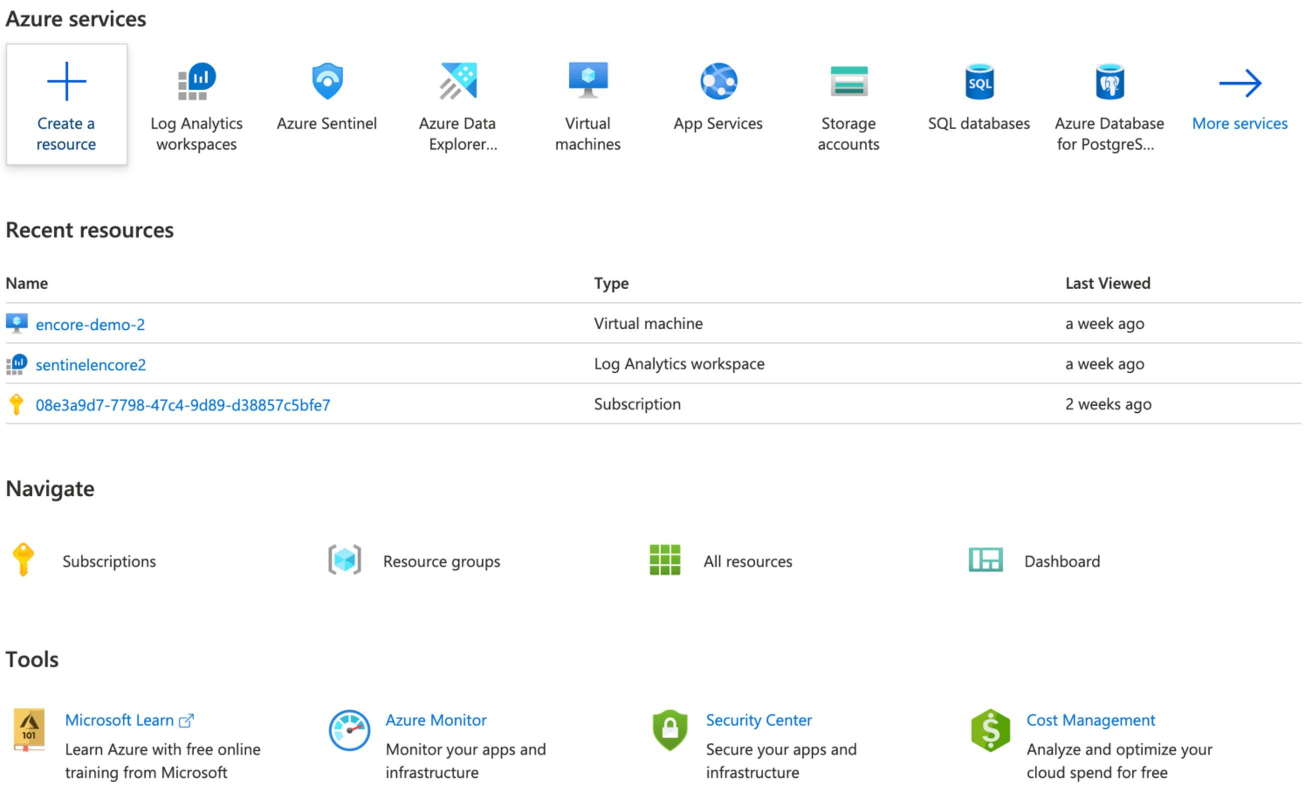
```
sudo pip3 install pyOpenSSL
```

2.3 EPEL Repo Dependency for RHEL

If you are having problems installing these packages, then you may need to enable the EPEL repository. Instructions for installing and enabling the EPEL repository are available on the World Wide Web.

2.4 Running eNcore CLI on Azure

Create a new Linux resource such as Ubuntu 18.04 LTS:



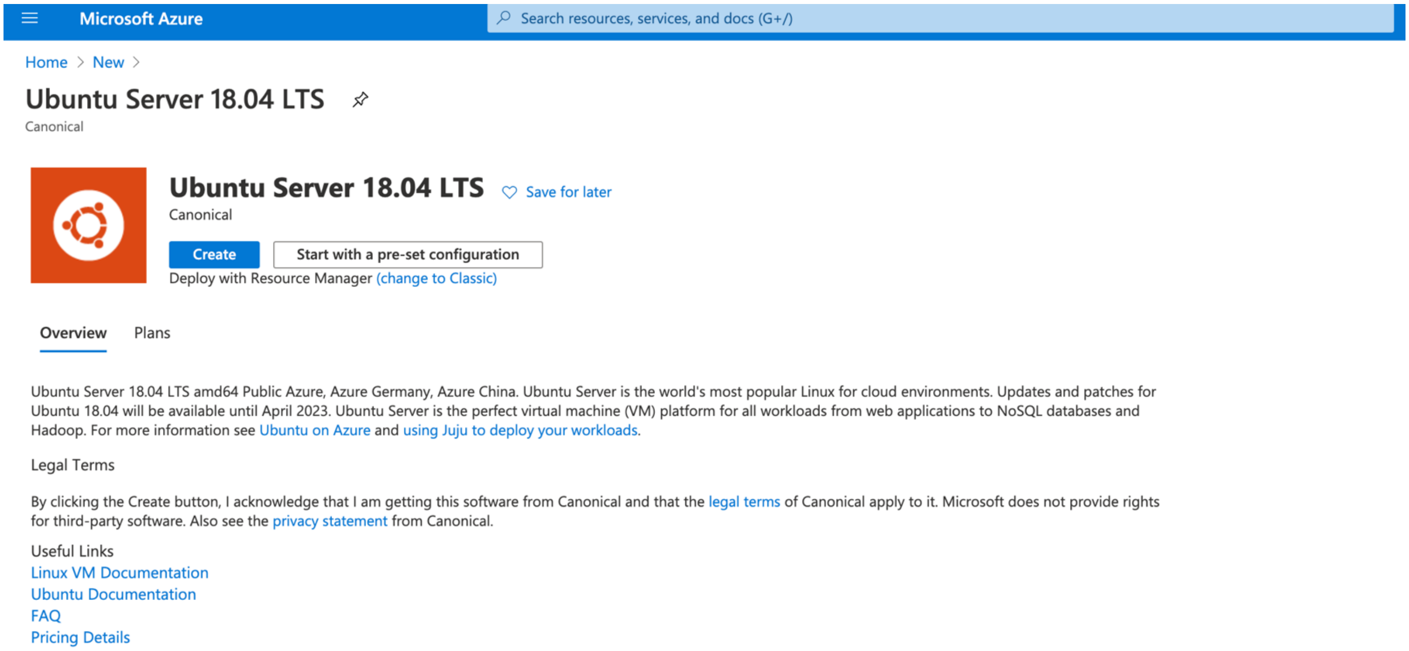
Name	Type	Last Viewed
encore-demo-2	Virtual machine	a week ago
sentinelencore2	Log Analytics workspace	a week ago
08e3a9d7-7798-47c4-9d89-d38857c5bfe7	Subscription	2 weeks ago

Microsoft Learn: Learn Azure with free online training from Microsoft

Azure Monitor: Monitor your apps and infrastructure

Security Center: Secure your apps and infrastructure

Cost Management: Analyze and optimize your cloud spend for free



Microsoft Azure
Search resources, services, and docs (G+/)
Home > New >
Save for later
Create
Start with a pre-set configuration
Overview
Plans

Ubuntu Server 18.04 LTS

Canonical

Ubuntu Server 18.04 LTS

Canonical

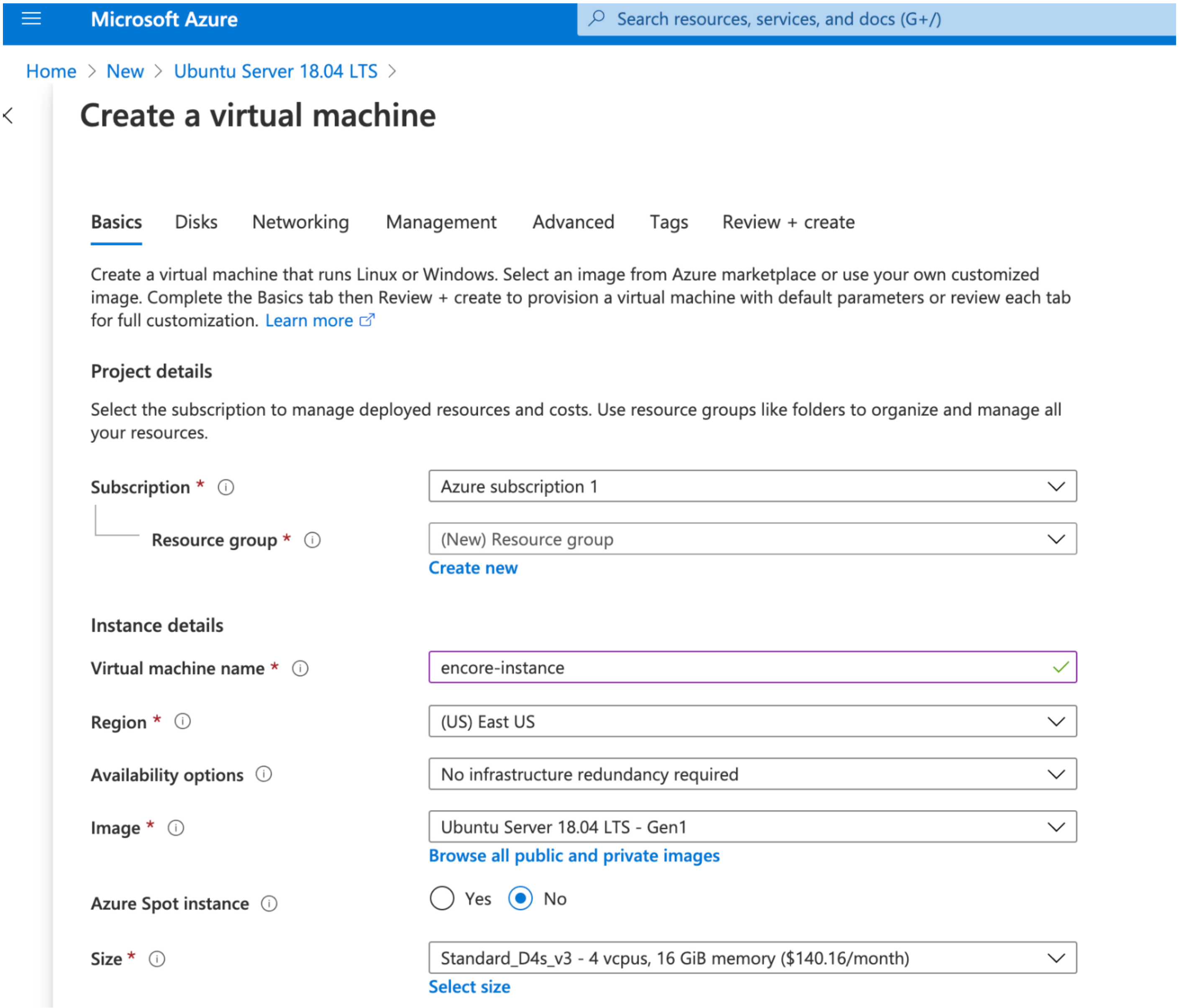
Deploy with Resource Manager (change to Classic)

Ubuntu Server 18.04 LTS amd64 Public Azure, Azure Germany, Azure China. Ubuntu Server is the world's most popular Linux for cloud environments. Updates and patches for Ubuntu 18.04 will be available until April 2023. Ubuntu Server is the perfect virtual machine (VM) platform for all workloads from web applications to NoSQL databases and Hadoop. For more information see Ubuntu on Azure and using Juju to deploy your workloads.

Legal Terms

By clicking the Create button, I acknowledge that I am getting this software from Canonical and that the legal terms of Canonical apply to it. Microsoft does not provide rights for third-party software. Also see the privacy statement from Canonical.

Useful Links
Linux VM Documentation
Ubuntu Documentation
FAQ
Pricing Details



Microsoft Azure
Search resources, services, and docs (G+/)
Home > New > Ubuntu Server 18.04 LTS >

Create a virtual machine

Basics
Disks
Networking
Management
Advanced
Tags
Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more

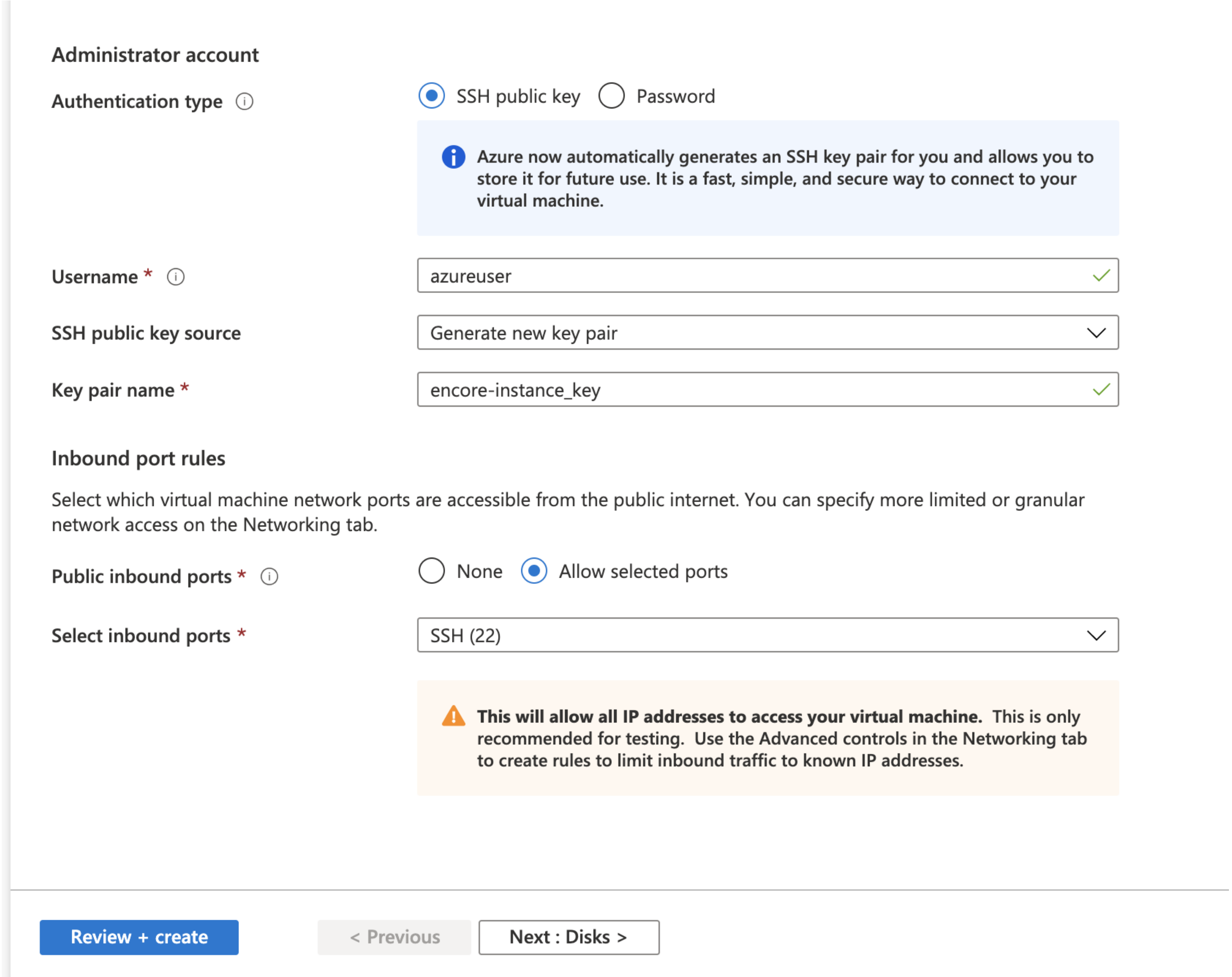
Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Azure subscription 1
Resource group *	(New) Resource group Create new

Instance details

Virtual machine name *	encore-instance ✓
Region *	(US) East US
Availability options	No infrastructure redundancy required
Image *	Ubuntu Server 18.04 LTS - Gen1 Browse all public and private images
Azure Spot instance	○ Yes ● No
Size *	Standard_D4s_v3 - 4 vcpus, 16 GiB memory ($140.16/month) Select size



Administrator account

Authentication type ⓘ ● SSH public key ○ Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username * ⓘ: azureuser ✓

SSH public key source: Generate new key pair

Key pair name *: encore-instance_key ✓

Inbound port rules

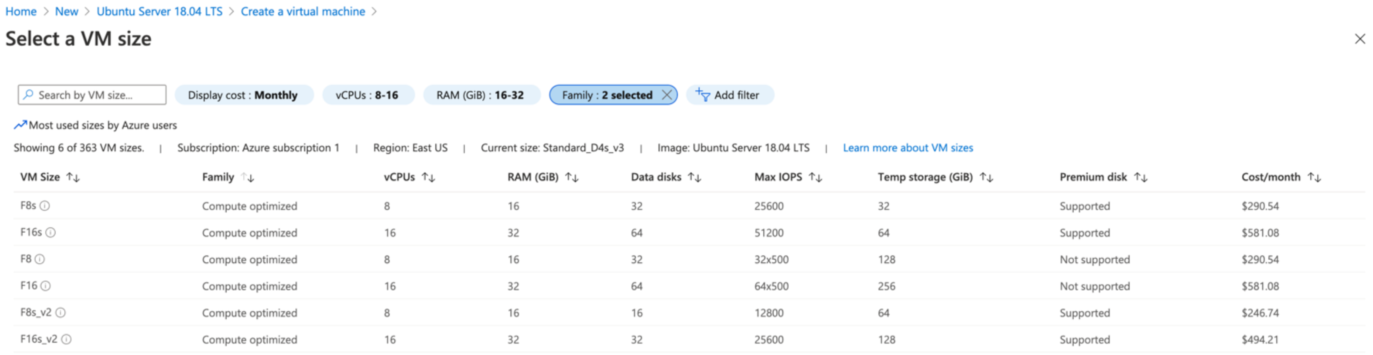
Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ ○ None ● Allow selected ports

Select inbound ports *: SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Review + create
< Previous
Next : Disks >



Home > New > Ubuntu Server 18.04 LTS > Create a virtual machine >

Select a VM size

Display cost : **Monthly** | vCPUs : **8-16** | RAM (GiB) : **16-32** | Family : **2 selected**

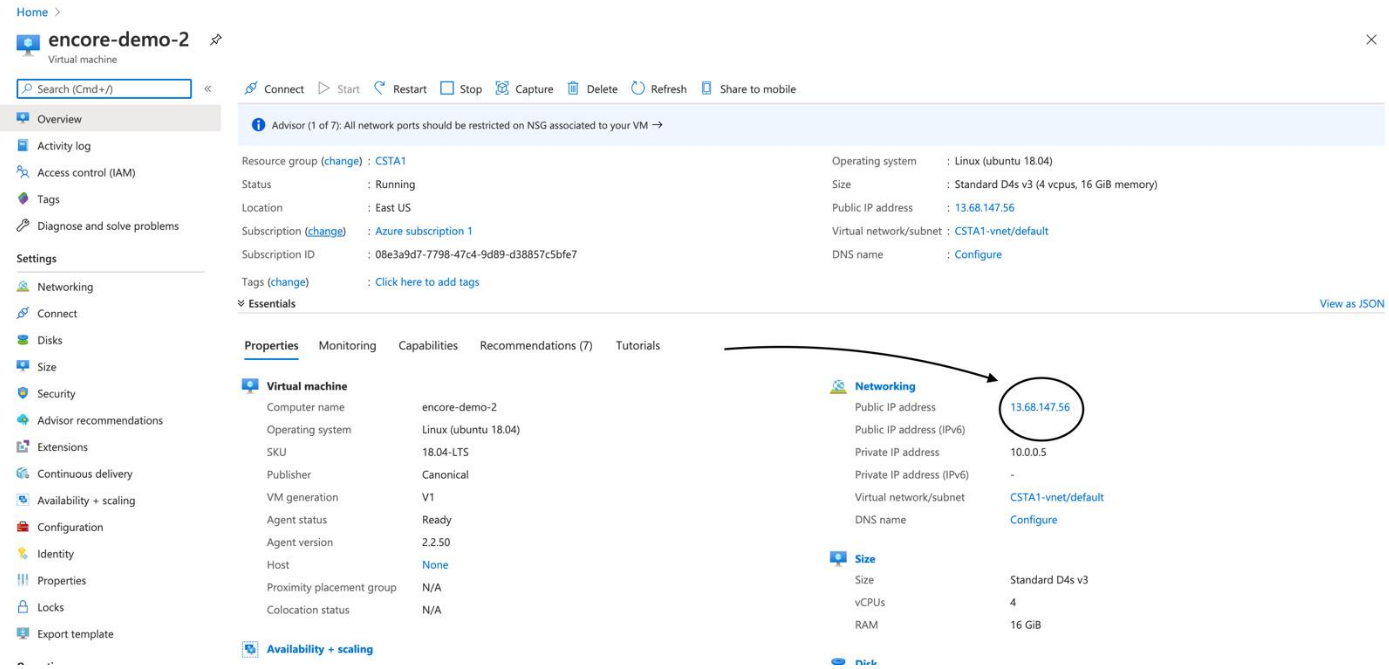
Add filter
Most used sizes by Azure users

Showing 6 of 363 VM sizes. | Subscription: Azure subscription 1 | Region: East US | Current size: Standard_D4s_v3 | Image: Ubuntu Server 18.04 LTS | Learn more about VM sizes

VM Size ↑↓	Family ↑↓	vCPUs ↑↓	RAM (GiB) ↑↓	Data disks ↑↓	Max IOPS ↑↓	Temp storage (GiB) ↑↓	Premium disk ↑↓	Cost/month ↑↓
F8s ⓘ	Compute optimized	8	16	32	25600	32	Supported	$290.54
F16s ⓘ	Compute optimized	16	32	64	51200	64	Supported	$581.08
F8 ⓘ	Compute optimized	8	16	32	32x500	128	Not supported	$290.54
F16 ⓘ	Compute optimized	16	32	64	64x500	256	Not supported	$581.08
F8s_v2 ⓘ	Compute optimized	8	16	16	12800	64	Supported	$246.74
F16s_v2 ⓘ	Compute optimized	16	32	32	25600	128	Supported	$494.21

Assign CPU(s) to the Virtual Instance. eNcore CLI can support up to 12 threads, we recommend 8-16 cores compute optimized, eNcore CLI can support up to 7k events/second using 16 CPU F16s_v2 option. Scale according to expected volume of your organization, the minimum recommended number of CPUs is 4 for low volume (>500 ev/sec) operations.

Name your instance and download the pem certificates



Advisor (1 of 7): All network ports should be restricted on NSG associated to your VM →

Resource group (change) : CSTA1
Status : Running
Location : East US
Subscription (change) : Azure subscription 1
Subscription ID : 08e3a9d7-7798-47c4-9d89-d38857c5bfe7
Tags (change) : Click here to add tags

Operating system : Linux (ubuntu 18.04)
Size : Standard D4s v3 (4 vcpus, 16 GiB memory)
Public IP address : 13.68.147.56
Virtual network/subnet : CSTA1-vnet/default
DNS name : Configure

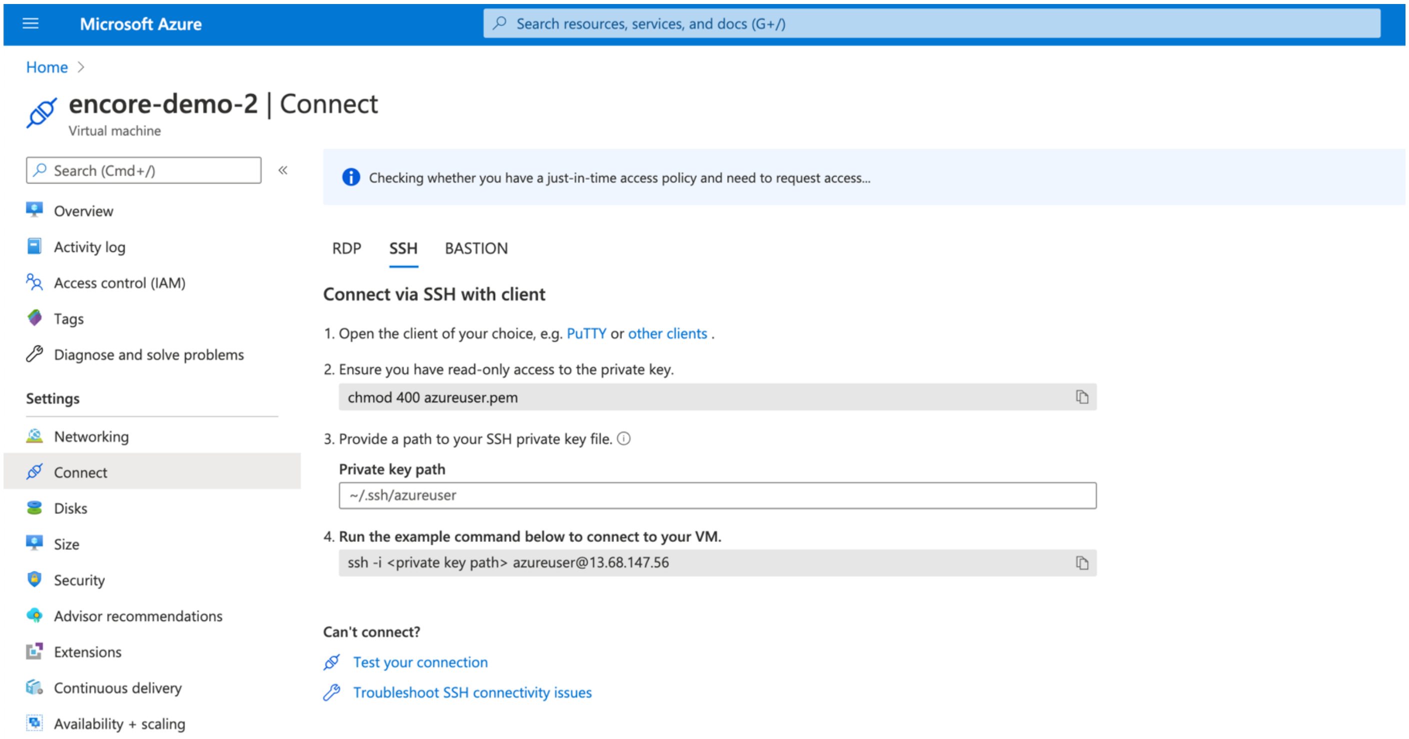
Virtual machine	
Computer name	encore-demo-2
Operating system	Linux (ubuntu 18.04)
SKU	18.04-LTS
Publisher	Canonical
VM generation	V1
Agent status	Ready
Agent version	2.2.50
Host	None
Proximity placement group	N/A
Colocation status	N/A

Networking	
Public IP address	13.68.147.56
Public IP address (IPv6)	
Private IP address	10.0.0.5
Private IP address (IPv6)	-
Virtual network/subnet	CSTA1-vnet/default
DNS name	Configure

Size	
Size	Standard D4s v3
vCPUs	4
RAM	16 GiB

Make note of the Public IP assigned to your instance, you will use this to create a certificate in the FMC eStreamer

Connect to the Command Line version of your instance using the pem file. Now you are ready to proceed with the installation. Azure also has a shortcut to enable a quick command line connection.



Checking whether you have a just-in-time access policy and need to request access...

Connect via SSH with client

1. Open the client of your choice, e.g. PuTTY or other clients .
2. Ensure you have read-only access to the private key.

```
chmod 400 azureuser.pem
```

3. Provide a path to your SSH private key file.

Private key path: ~/.ssh/azureuser

4. Run the example command below to connect to your VM.

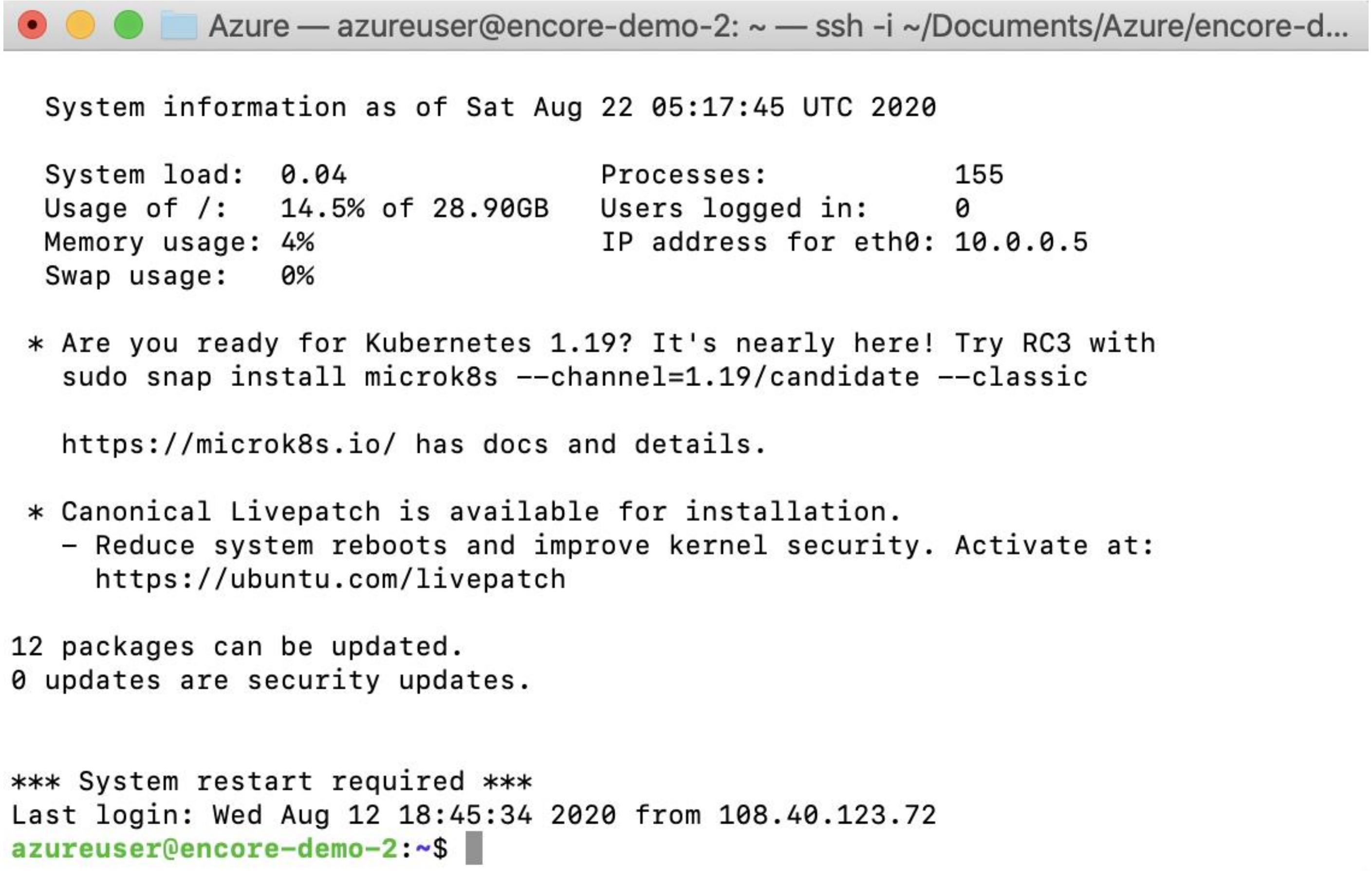
```
ssh -i <private key path> azureuser@13.68.147.56
```

Can't connect?

Test your connection

Troubleshoot SSH connectivity issues

```
ssh -I <private key path> azureuser@<public ip>
```


Azure — azureuser@encore-demo-2: ~ — ssh -i ~/Documents/Azure/encore-d...

```
  System information as of Sat Aug 22 05:17:45 UTC 2020

  System load:  0.04               Processes:           155
  Usage of /:   14.5% of 28.90GB   Users logged in:     0
  Memory usage: 4%                 IP address for eth0: 10.0.0.5
  Swap usage:   0%

 * Are you ready for Kubernetes 1.19? It's nearly here! Try RC3 with
   sudo snap install microk8s --channel=1.19/candidate --classic

   https://microk8s.io/ has docs and details.

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

12 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Wed Aug 12 18:45:34 2020 from 108.40.123.72
azureuser@encore-demo-2:~$
```

2.5 Running eNcore CLI on Windows

Warning: Windows is not yet supported for production execution. If, however, you wish to attempt an install for the CLI version, then you will need to run the following commands.

```
pip install pyOpenSSL
pip install win-inet-pton
```

3 Installing eStreamer eNcore CLI

3.1 Build the eNcore Client from Source

Use the following command to copy the file from your local machine to the target device:

git clone https://github.com/CiscoSecurity/fp-05-microsoft-sentinel-connector.git

The project can also be downloaded to zip or

3.3 Create (or copy existing) PKCS12 file

See Appendix A for instructions on how to create a PKCS12 file in the FMC and download it.

3.4 Install the PKCS12 File

Use the following command to securely copy the pkcs12 file to the eNcore CLI installation.

```
scp -i /path/to/pem/encore-demo-2_key.pem /local/path/<public ip>.pkcs12 azureuser@<Public     Ip>:/tmp/
```

Copy the certificate from /tmp to the runtime path of the git project

```
cp /tmp/client.pkcs12 ~/fp-05-microsoft-sentinel-connector
```

3.6.8 Test

Change the working directory to /using the following command:

```
cd ~/fp-05-microsoft-sentinel-connector
```

Then, run the encore shell script – you will be guided through any additional configuration:

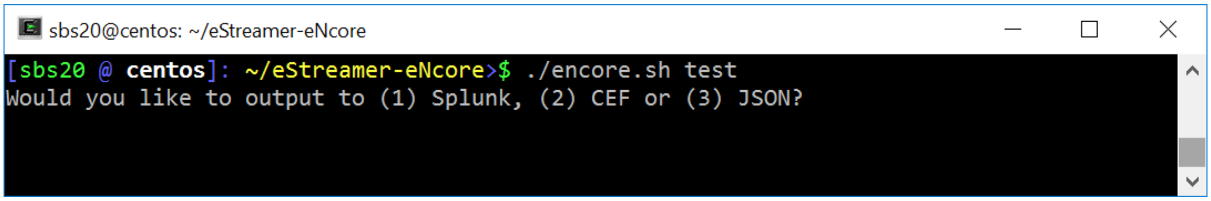
./encore.sh test

The script will verify that you have the pre-requisites installed, notably:

- Python 2.7, Python 3.6+ requires “python3” branch from git
- the correct build of Python
- pyOpenSSL
- a client.pkcs12 file
- a valid host
- It will prompt you to choose whether to output data for Splunk, CEF or JSON, in this guide we use the CEF outputter, however future versions may use JSON or other custom formats on depending on the Sentinel Connector being used

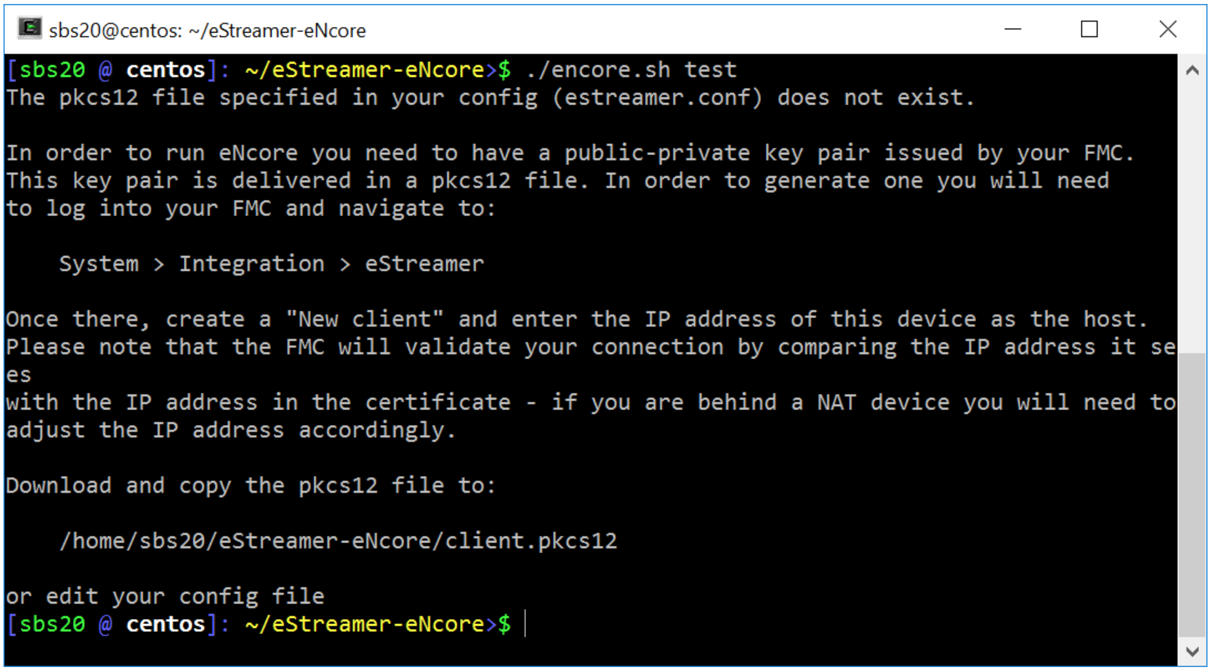
If there are any missing items, you will be presented with an explanation. An example explanation is in the following figure.

Figure 1. Choosing your output



```
[sbs20 @ centos]: ~/eStreamer-eNcore>$ ./encore.sh test
Would you like to output to (1) Splunk, (2) CEF or (3) JSON?
```

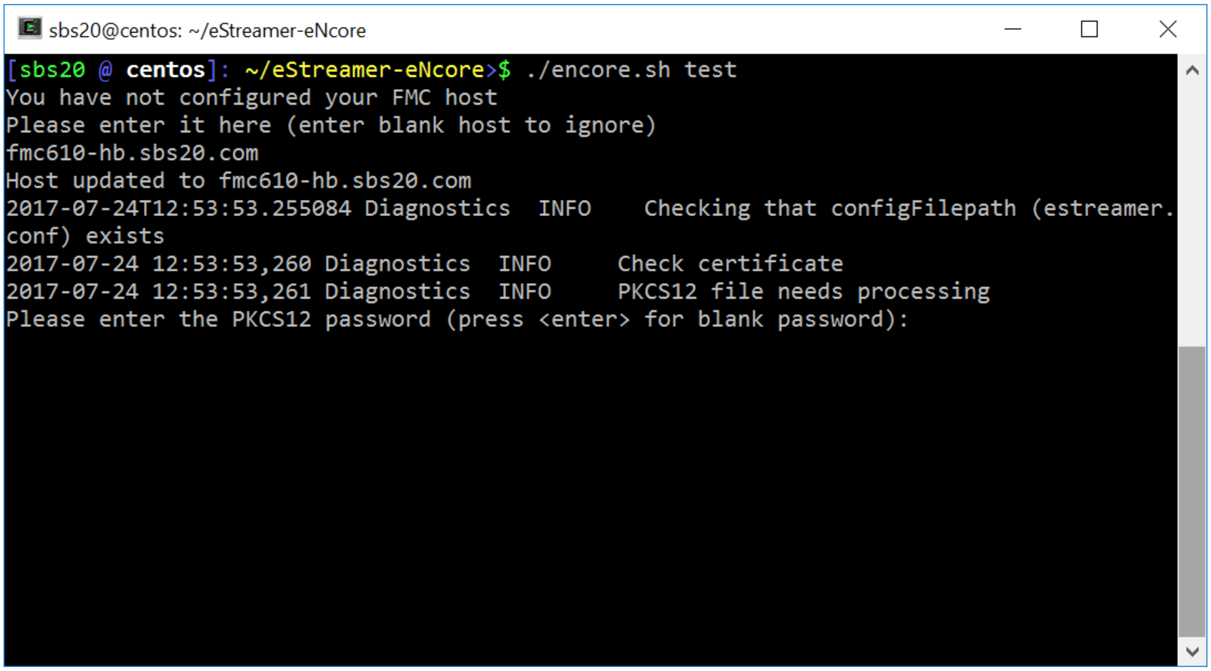
Figure 2: Missing pkcs12 File



```
[sbs20 @ centos]: ~/eStreamer-eNcore>$ ./encore.sh test
The pkcs12 file specified in your config (estreamer.conf) does not exist.

In order to run eNcore you need to have a public-private key pair issued by your FMC.
This key pair is delivered in a pkcs12 file. In order to generate one you will need
to log into your FMC and navigate to:

    System > Integration > eStreamer

Once there, create a "New client" and enter the IP address of this device as the host.
Please note that the FMC will validate your connection by comparing the IP address it se
es
with the IP address in the certificate - if you are behind a NAT device you will need to
adjust the IP address accordingly.

Download and copy the pkcs12 file to:

    /home/sbs20/eStreamer-eNcore/client.pkcs12

or edit your config file
[sbs20 @ centos]: ~/eStreamer-eNcore>$
```

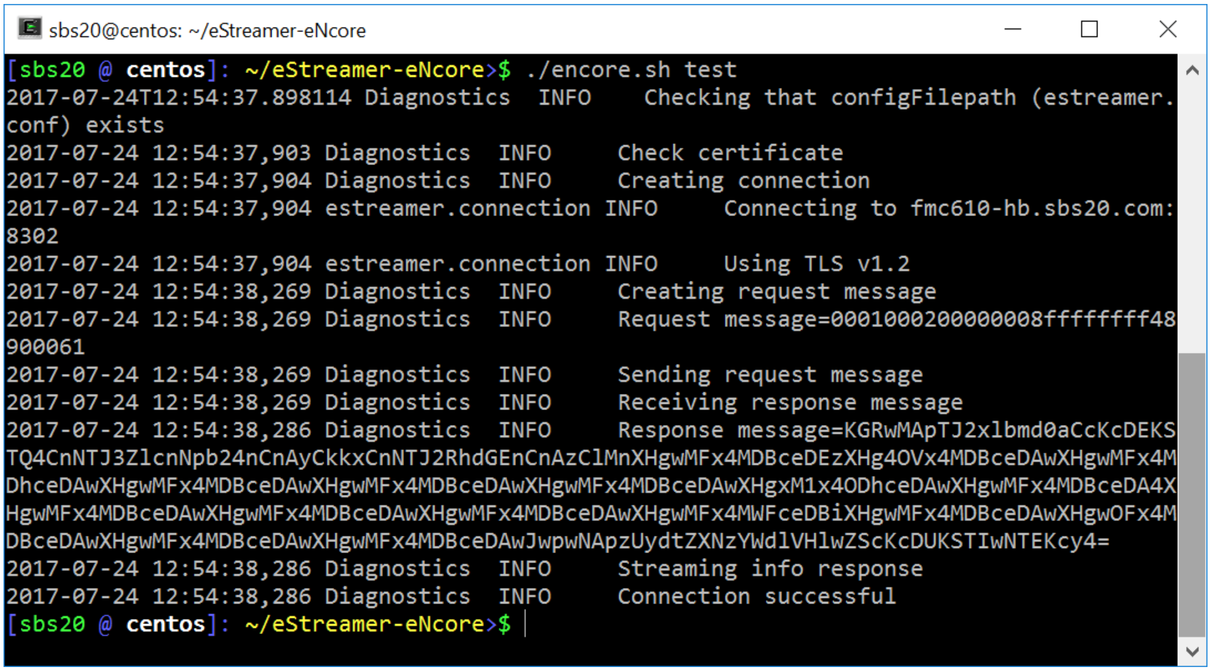
You will then be prompted to enter the IP / FQDN of the FMC and the PKCS12 file password.

Figure 3: Enter Password



```
[sbs20 @ centos]: ~/eStreamer-eNcore>$ ./encore.sh test
You have not configured your FMC host
Please enter it here (enter blank host to ignore)
fmc610-hb.sbs20.com
Host updated to fmc610-hb.sbs20.com
2017-07-24T12:53:53.255084 Diagnostics  INFO    Checking that configFilepath (estreamer.
conf) exists
2017-07-24 12:53:53,260 Diagnostics  INFO     Check certificate
2017-07-24 12:53:53,261 Diagnostics  INFO     PKCS12 file needs processing
Please enter the PKCS12 password (press <enter> for blank password):
```

Figure 4: Successful Test

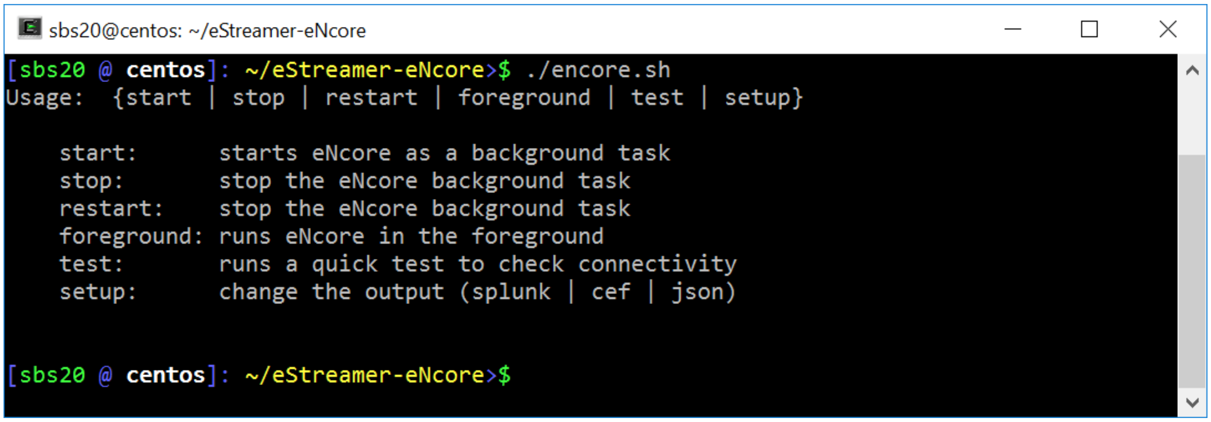


```
[sbs20 @ centos]: ~/eStreamer-eNcore>$ ./encore.sh test
2017-07-24T12:54:37.898114 Diagnostics  INFO    Checking that configFilepath (estreamer.
conf) exists
2017-07-24 12:54:37,903 Diagnostics  INFO     Check certificate
2017-07-24 12:54:37,904 Diagnostics  INFO     Creating connection
2017-07-24 12:54:37,904 estreamer.connection INFO     Connecting to fmc610-hb.sbs20.com:
8302
2017-07-24 12:54:37,904 estreamer.connection INFO     Using TLS v1.2
2017-07-24 12:54:38,269 Diagnostics  INFO     Creating request message
2017-07-24 12:54:38,269 Diagnostics  INFO     Request message=0001000200000008ffffffff48
900061
2017-07-24 12:54:38,269 Diagnostics  INFO     Sending request message
2017-07-24 12:54:38,269 Diagnostics  INFO     Receiving response message
2017-07-24 12:54:38,286 Diagnostics  INFO     Response message=KGRwMApTJ2xlbmd0aCcKcDEKS
TQ4CnNTJ3ZlcnNpb24nCnAyCkkxCnNTJ2RhdGEnCnAzClMnXHgwMFx4MDBceDEzXHg4OVx4MDBceDAwXHgwMFx4M
DhceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgxM1x4ODhceDAwXHgwMFx4MDBceDA4X
HgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MDBceDAwXHgwMFx4MWFceDBiXHgwMFx4MDBceDAwXHgwOFx4M
DBceDAwXHgwMFx4MDBceDAwJwpwNApzUydtZXNzYWdlVHlwZScKcDUKSTIwNTEKcy4=
2017-07-24 12:54:38,286 Diagnostics  INFO     Streaming info response
2017-07-24 12:54:38,286 Diagnostics  INFO     Connection successful
[sbs20 @ centos]: ~/eStreamer-eNcore>$
```

4. Running eNcore CLI

If you run encore.sh without any parameters, you will be presented with brief instructions.

Figure 5: Help Screen

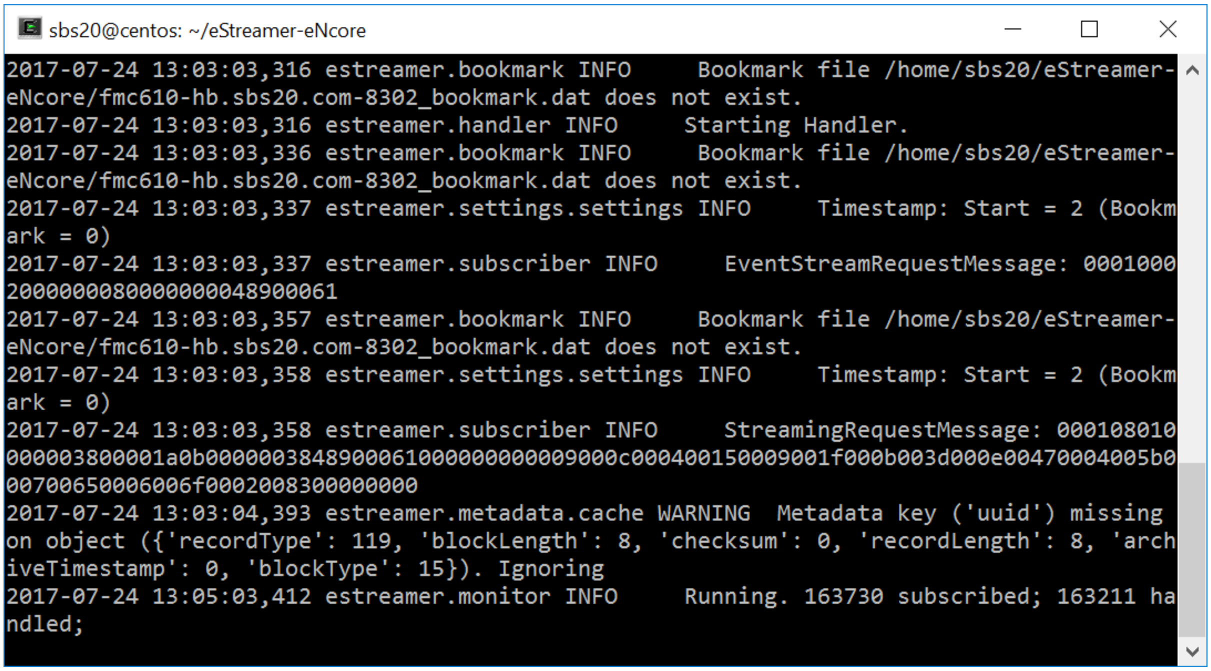


sbs20@centos: ~/eStreamer-eNcore

```
[sbs20 @ centos]: ~/eStreamer-eNcore>$ ./encore.sh
Usage:  {start | stop | restart | foreground | test | setup}

    start:      starts eNcore as a background task
    stop:       stop the eNcore background task
    restart:    stop the eNcore background task
    foreground: runs eNcore in the foreground
    test:       runs a quick test to check connectivity
    setup:      change the output (splunk | cef | json)


[sbs20 @ centos]: ~/eStreamer-eNcore>$
```

For your first run, run it in the foreground so you can see what is happening. Every two minutes, the screen will update with a note of how many records have been processed. If you wish to change the update frequency, see the monitor.period configuration setting.

Figure 6: Running in the Foreground with Monitor Status



sbs20@centos: ~/eStreamer-eNcore

```
2017-07-24 13:03:03,316 estreamer.bookmark INFO     Bookmark file /home/sbs20/eStreamer-
eNcore/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-07-24 13:03:03,316 estreamer.handler INFO     Starting Handler.
2017-07-24 13:03:03,336 estreamer.bookmark INFO     Bookmark file /home/sbs20/eStreamer-
eNcore/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-07-24 13:03:03,337 estreamer.settings.settings INFO     Timestamp: Start = 2 (Bookm
ark = 0)
2017-07-24 13:03:03,337 estreamer.subscriber INFO     EventStreamRequestMessage: 0001000
20000000800000000048900061
2017-07-24 13:03:03,357 estreamer.bookmark INFO     Bookmark file /home/sbs20/eStreamer-
eNcore/fmc610-hb.sbs20.com-8302_bookmark.dat does not exist.
2017-07-24 13:03:03,358 estreamer.settings.settings INFO     Timestamp: Start = 2 (Bookm
ark = 0)
2017-07-24 13:03:03,358 estreamer.subscriber INFO     StreamingRequestMessage: 000108010
000003800001a0b00000038489000610000000000009000c000400150009001f000b003d000e00470004005b0
00700650006006f0002008300000000
2017-07-24 13:03:04,393 estreamer.metadata.cache WARNING  Metadata key ('uuid') missing
on object ({'recordType': 119, 'blockLength': 8, 'checksum': 0, 'recordLength': 8, 'arch
iveTimestamp': 0, 'blockType': 15}). Ignoring
2017-07-24 13:05:03,412 estreamer.monitor INFO     Running. 163730 subscribed; 163211 ha
ndled;
```

Note: To stop the foreground process, press ctrl-c.

5. Configuration Options

5.1 Essential Configuration

The default configuration file is set up to run out of the box. Following is a brief explanation of each setting in case you wish to customize.

5.1.1 Subscription Server

This is the FMC host and associated information. If you encounter TLS difficulties and are willing to downgrade, then you can change tlsVersion to 1.0.

Note: Note that downgrading the TLS version is useful for debugging and seeing the software work but it is not a recommended long-term strategy. It is recommended instead to fix the root cause.

Figure 8: Subscription Server Screen

```
"subscription": {
  "servers": [
    {
      "host": "1.2.3.4",
      "port": 8302,
      "pkcs12Filepath": "client.pkcs12",
      "@comment": "Valid values are 1.0 and 1.2",
      "tlsVersion": 1.2
    }
  ], ...
```

5.1.2 Monitor

The monitor is a separate thread that runs monitoring and maintenance tasks. By default, it runs every two minutes. It will report the number of events received and handled and will check the status of sub-processes. If there have been any problems, the monitor will place the client into an error state and the client will shut itself down.

Figure 9: Monitor Screen

```
"monitor": {
  "period": 120,
  "velocity": false,
  "bookmark": false,
  "subscribed": true,
  "handled": true
},
```

5.1.3 Start

The eStreamer server expects requests to state their chosen start time. There are broadly three options:

- 0: Return all data from the earliest point available on the FMC
- 1: Return all data from now onwards
- 2: Use a bookmark to pick up where we left off. First run is from 0

Figure 10: Start Screen

```
"@startComment": "0 for genesis, 1 for now, 2 for bookmark",
"start": 2,
```

5.14 Outputters (Output Data Location)

Two examples of outputters are given in the figure below. Although only one outputter is required – one that sends CEF events to the Sentinel connector, it is often useful to write CEF output to local files. The second outputter shown in the figure below writes the CEF events to local files.

Figure 11: Outputters Screen

```
"outputters": [
  {
    "name": "CEF",
    "adapter": "cef",
    "enabled": true,
    "stream": {
      "uri": "udp://10.0.1.2:514",
    }
  },

  {
    "name": "CEFfile",
    "adapter": "cef",
    "enabled": true,
    "stream": {
      "uri": "relfile:///data/data.{0}.cef",
      "options": {
        "rotate": false,
        "maxLogs": 9999
      }
    }
  }

]
```

5.2 Advanced Configuration Options

Key	Definition

Key	Definition
alwaysAttemptToContinue	true | false. Controls whether eNcore client will persist a connection even if the CLI process has been terminated
Enabled	true | false. Controls whether eNcore will run.
connectTimeout	The duration in seconds the client will wait for a connection to establish before failing.
responseTimeout	The duration in seconds the client will wait for a response before timing out.
monitor.period	The period in seconds between each execution of monitor tasks. Default is 120. Lower numbers are useful for debugging but will create more log traffic.
monitor.velocity	true | false. True will display the speed at which the client is processing records. A positive value means the client is processing events faster than eStreamer is sending them. Negative is slower. Once up to date, this should hover around zero.
monitor.bookmark	true | false. True will show the last bookmark timestamp. This is useful to see how far behind the eNcore client is.
monitor.subscribed	true | false. True will report the total number of events subscribed.
monitor.handled	true | false. True will report the total number of events written to output.
Start	0 specifies oldest data available 1 specifies data as of now 2 specifies use of bookmark
logging.level	Levels include FATAL, ERROR, WARNING, INFO, DEBUG, VERBOSE, and TRACE. Select the level of logging as per your requirement. It is strongly recommended that you do not use anything above INFO for production environments. DEBUG will generate very large log files and TRACE will significantly affect performance.
logging.format	This describes the format of the log and how they are stored. Default configuration setting for message format is "{date-time}-{name of module}-{level of logging-message}".
logging.stdOut	true | false. This determines whether log output is also shown in Standard Output.
logging.filepath	This specifies the location of the application log.
maxQueueSize	Maximum number of messages buffered before throttling takes place. It is essentially a buffer size. The larger this number, the longer it will take to shutdown. Default configuration setting is 100. Do not change.

Key	Definition
subscription.servers[]	While this is an array, eNcore can only currently support one server. The array is to support the future ability to connect to multiple hosts.
server.host	The IP address of the FMC (eStreamer Server). Default configuration is 1.2.3.4. If you change the host entry after having run eNcore then new cache, bookmark and metadata files will be generated.
server.port	The server port to connect to. Default 8302.
server.pkcs12Filepath	The PKCS12 filepath location. If you change this having already run eNcore, then you must also delete the cached public and private key otherwise eNcore will continue to use those. They are called {host}-{port}_pkcs.cert and {host}-{port}_pkcs.key.
server.tlsVersion	Valid options are 1.0 and 1.2.
subscription.records	Do not change these values.
handler.records.metadata	true | false. If you wish to exclude the output of metadata (since it has no timestamp information) then set this to false.
handler.records.flows	true | false. If you wish to exclude connection flow records then set this to false.
handler.outputters[]	An array of outputter controllers which define the behavior and format of what gets written by eNcore.
outputter.name	This is a human readable name for your conveience. It is unused by the code.
outputter.adapter	Data is read from eStreamer and stored in a structured internal format. The adapter transforms the data to a desired format. Recognized values are: — splunk — json
outputter.enabled	true | false. You can have more than one outputter specified at once. If you wish to disable a specific outputter, set this flag to false. If all outputters are false (or there are no outputters) then it behaves as a sink.
outputter.passthru	true | false. If true then data flowing through bypasses decoding and metadata processing. It is very fast but of limited use. Its primary purpose is for debugging.
outputter.stream.uri	Specify the location where the output will be stored. You can specify a file URI as normal (e.g., file:///absolute/path/to/file) or a relative filepath (relfile:////relative/path/to/file). Only file URLs are supported currently.
outputter.stream.options	File-based streams require additional options.

Key	Definition
option.rotate	true | false. Set if you want log rotation. Default configuration setting for this is true. Please note that eNcore will not delete any old files. If you wish to do that, you will need to script it separately and schedule it. Example: Call this from a cron job. #!/bin/bash find /opt/splunk/etc/apps/eStreamer/log/* -mmin +1440 -exec rm {} \;
option.maxLogs	Specify the size of the log (number of lines). *Default configuration for this is 10,000. You can have fewer, larger files (e.g, 50,000).*

5.3 Execution

Various shell scripts options are available.

During installation and initial setup – or perhaps for debugging purposes it is useful to run the following commands.

```
./encore.sh test
```

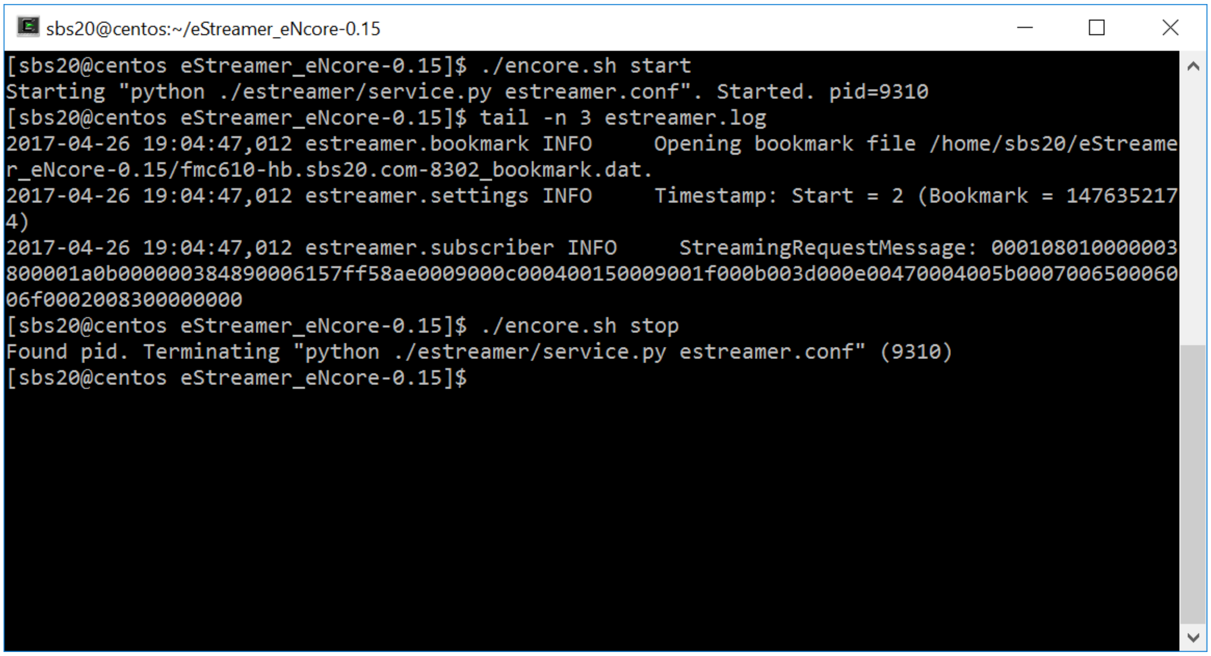
And

```
./encore.sh foreground
```

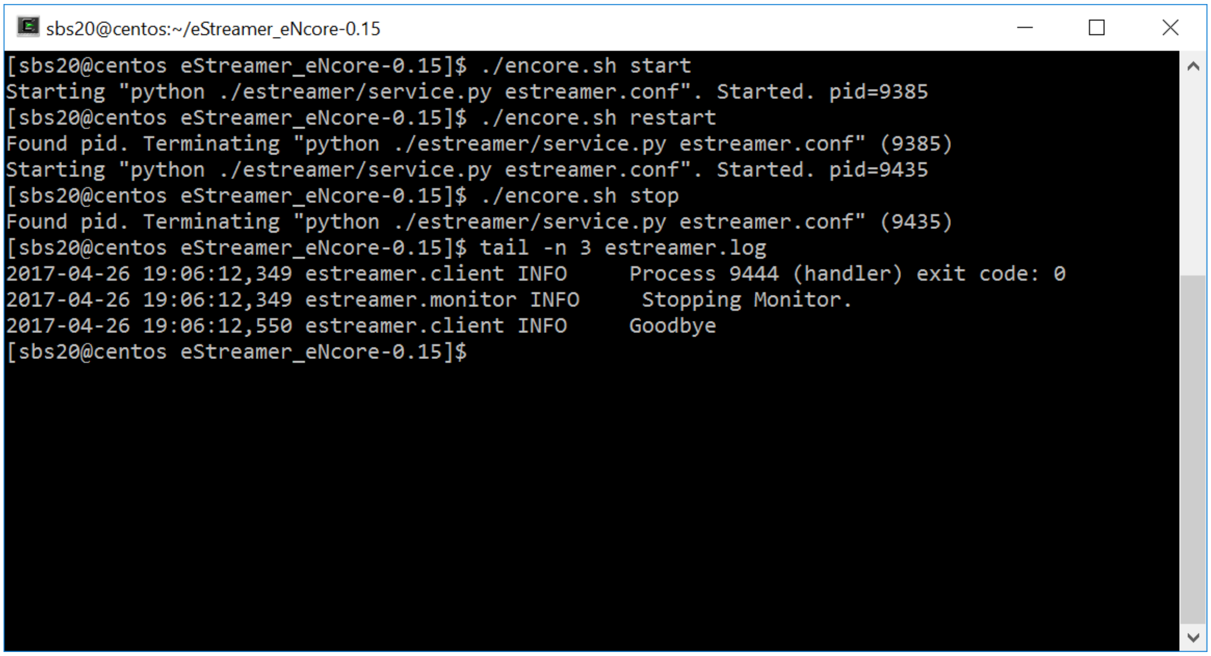
In all other cases, it is expected that encore will be run in the background, for which the following commands are pertinent.

```
./encore.sh start
./encore.sh stop
./encore.sh restart
```

Figure 12: Start, Tail Log, Stop



```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh start
Starting "python ./estreamer/service.py estreamer.conf". Started. pid=9310
[sbs20@centos eStreamer_eNcore-0.15]$ tail -n 3 estreamer.log
2017-04-26 19:04:47,012 estreamer.bookmark INFO     Opening bookmark file /home/sbs20/eStreame
r_eNcore-0.15/fmc610-hb.sbs20.com-8302_bookmark.dat.
2017-04-26 19:04:47,012 estreamer.settings INFO     Timestamp: Start = 2 (Bookmark = 147635217
4)
2017-04-26 19:04:47,012 estreamer.subscriber INFO     StreamingRequestMessage: 000108010000003
800001a0b000000384890006157ff58ae0009000c000400150009001f000b003d000e00470004005b000700650006
06f0002008300000000
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh stop
Found pid. Terminating "python ./estreamer/service.py estreamer.conf" (9310)
[sbs20@centos eStreamer_eNcore-0.15]$
```



```
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh start
Starting "python ./estreamer/service.py estreamer.conf". Started. pid=9385
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh restart
Found pid. Terminating "python ./estreamer/service.py estreamer.conf" (9385)
Starting "python ./estreamer/service.py estreamer.conf". Started. pid=9435
[sbs20@centos eStreamer_eNcore-0.15]$ ./encore.sh stop
Found pid. Terminating "python ./estreamer/service.py estreamer.conf" (9435)
[sbs20@centos eStreamer_eNcore-0.15]$ tail -n 3 estreamer.log
2017-04-26 19:06:12,349 estreamer.client INFO     Process 9444 (handler) exit code: 0
2017-04-26 19:06:12,349 estreamer.monitor INFO     Stopping Monitor.
2017-04-26 19:06:12,550 estreamer.client INFO     Goodbye
[sbs20@centos eStreamer_eNcore-0.15]$
```

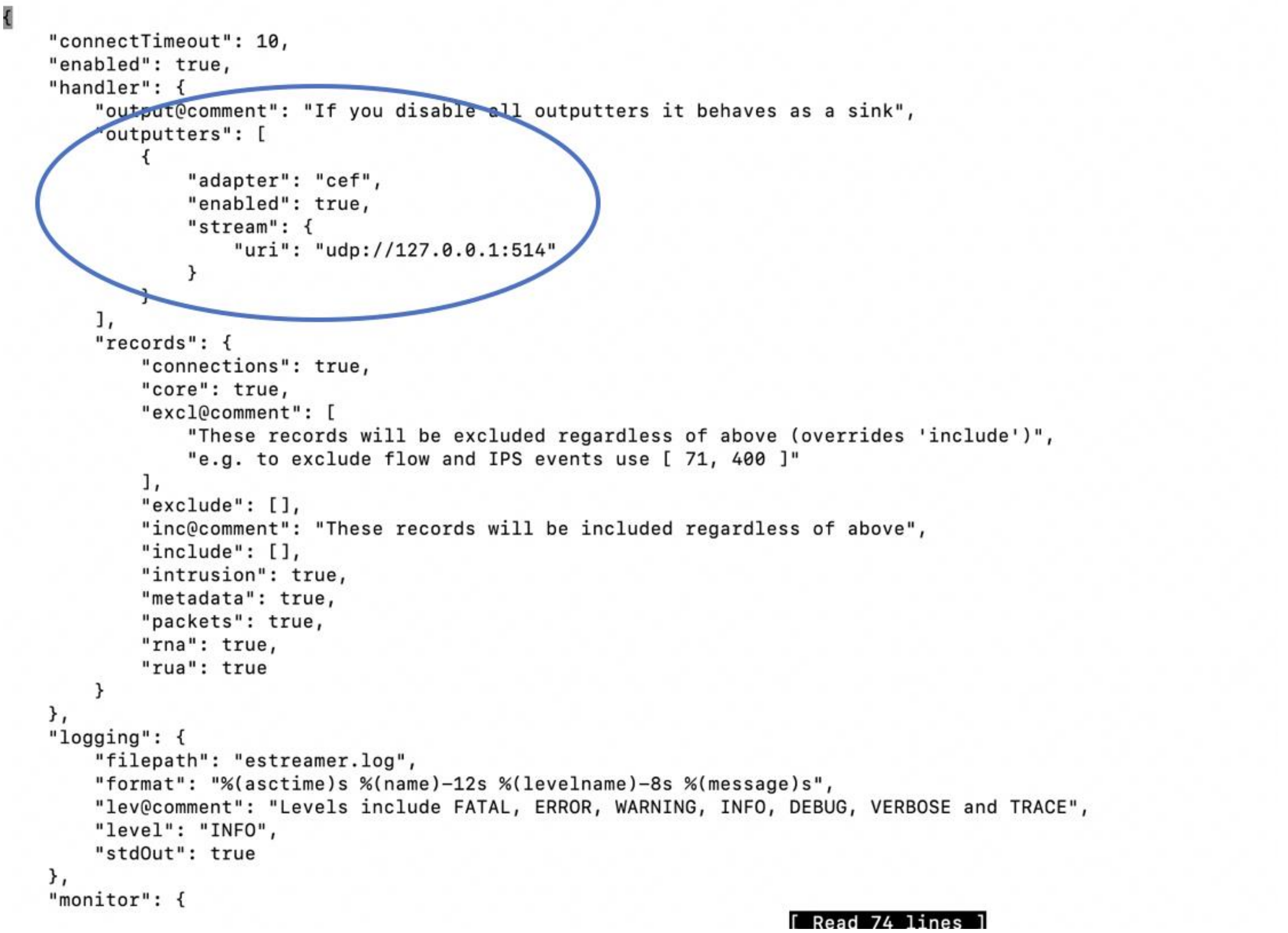
5.4 Logging

By default, eNcore will output an estreamer.log application log in its working directory with a log level of INFO. The format of the log file can be adjusted using the logging.format configuration setting. The level can also be adjusted. It is recommended that the default settings are left in place for production execution.

6 Sending data to Sentinel

6.1 Configuring Encore to Stream UDP

Configure encore to stream CEF data using UDP on port 514



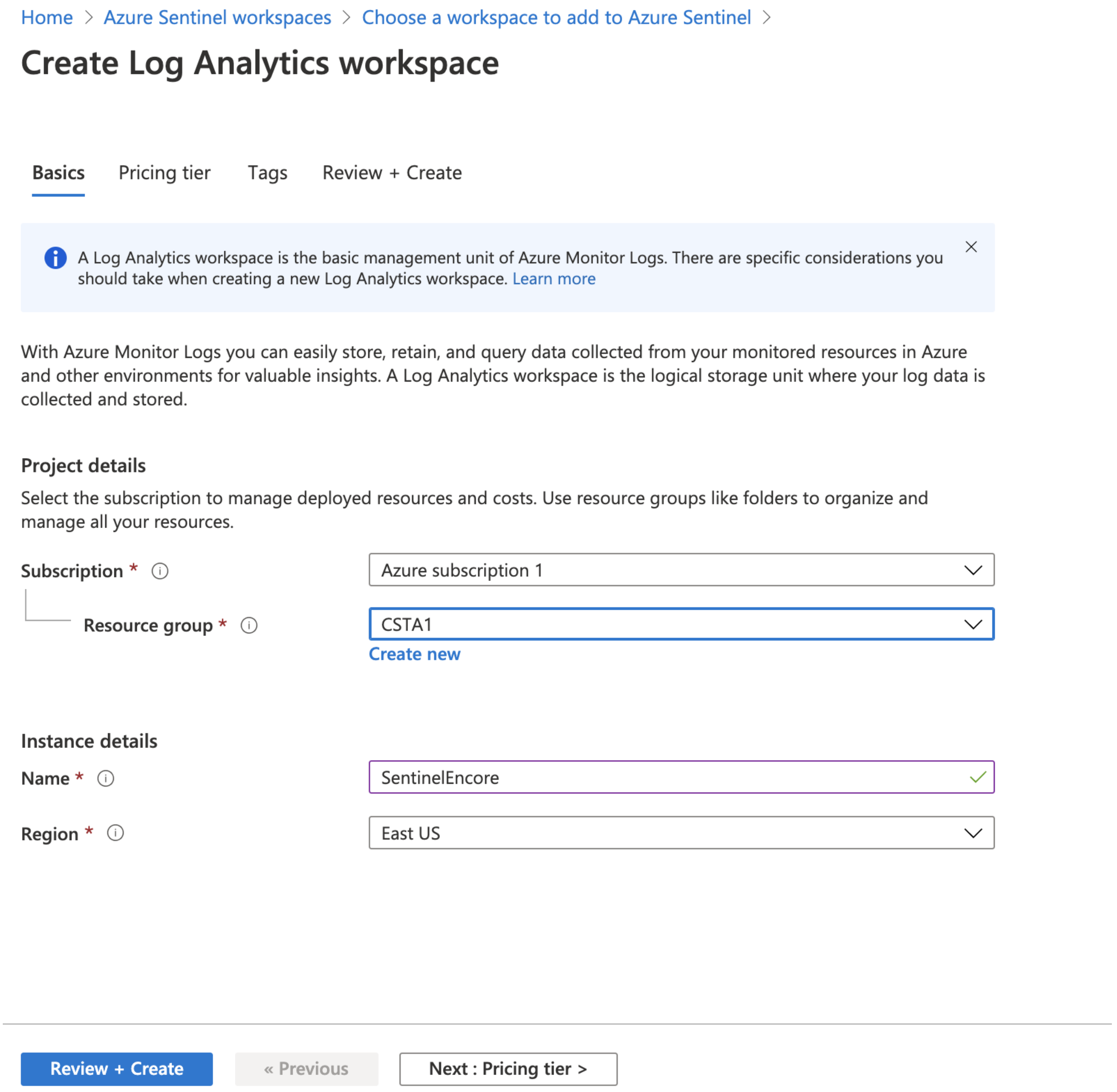
```
{
    "connectTimeout": 10,
    "enabled": true,
    "handler": {
        "output@comment": "If you disable all outputters it behaves as a sink",
        "outputters": [
            {
                "adapter": "cef",
                "enabled": true,
                "stream": {
                    "uri": "udp://127.0.0.1:514"
                }
            }
        ],
        "records": {
            "connections": true,
            "core": true,
            "excl@comment": [
                "These records will be excluded regardless of above (overrides 'include')",
                "e.g. to exclude flow and IPS events use [ 71, 400 ]"
            ],
            "exclude": [],
            "inc@comment": "These records will be included regardless of above",
            "include": [],
            "intrusion": true,
            "metadata": true,
            "packets": true,
            "rna": true,
            "rua": true
        }
    },
    "logging": {
        "filepath": "estreamer.log",
        "format": "%(asctime)s %(name)-12s %(levelname)-8s %(message)s",
        "lev@comment": "Levels include FATAL, ERROR, WARNING, INFO, DEBUG, VERBOSE and TRACE",
        "level": "INFO",
        "stdOut": true
    },
    "monitor": {
                                                                    [ Read 74 lines ]
```

If encore is already in process use the encore.sh stop/start command to restart encore

6.2 Creating a Sentinel Workspace

Once you’ve established a working eNcore client between the FMC and your Azure instance you can route your data outputs to Sentinel using an agent collector

If you don't have a Sentinel Workspace proceed with the following.



Home > Azure Sentinel workspaces > Choose a workspace to add to Azure Sentinel >

Create Log Analytics workspace

A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. Learn more

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure subscription 1

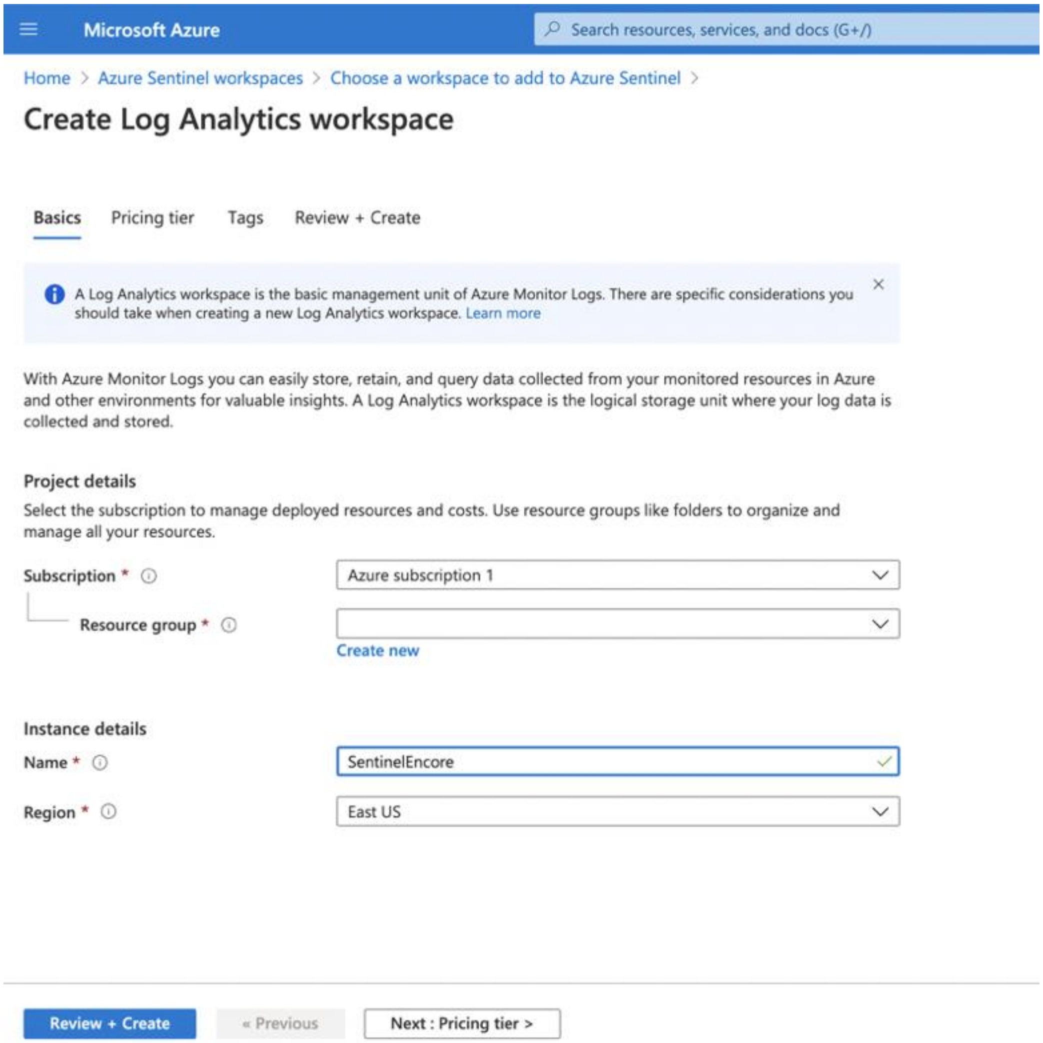
Resource group * : CSTA1

Create new

Instance details

Name * : SentinelEncore

Region * : East US


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Azure Sentinel workspaces > Choose a workspace to add to Azure Sentinel >
Basics
Pricing tier
Tags
Review + Create
Review + Create
« Previous
Next : Pricing tier >

Create Log Analytics workspace

A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. Learn more

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Azure subscription 1

Resource group *:
Create new

Instance details

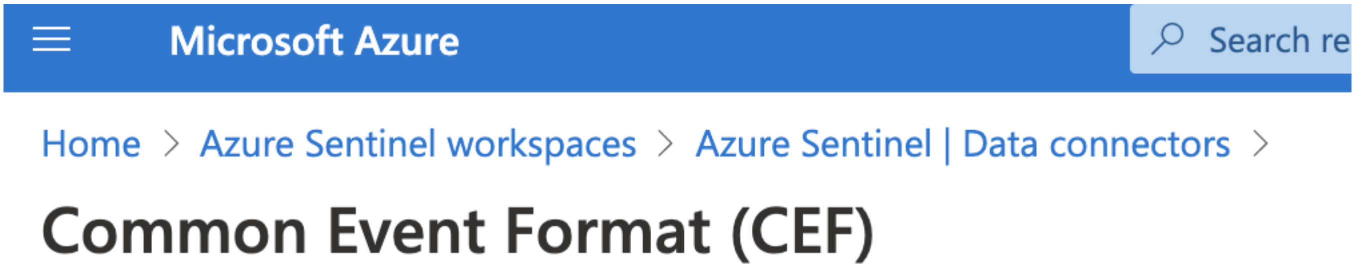
Name *: SentinelEncore

Region *: East US

6.2 Setting up the CEF Data Connector

Once you've established a working eNcore client between the FMC and your Azure instance you can route your data outputs to Sentinel using an agent collector

Please refer the official Microsoft guide (https://docs.microsoft.com/en-us/azure/sentinel/connect-cef-agent?tabs=rsyslog) , accessing


Microsoft Azure
Search re
Home > Azure Sentinel workspaces > Azure Sentinel | Data connectors >

Common Event Format (CEF)

Accessing the connector documentation guide directly from Sentinel is preferred as the docs and prepopulated commands will contain workspace and primary key information specific to your Azure instance. The following steps below are directly from the Azure Sentinel setup guide for reference, again it is better to use direct documentation with the Sentinel platform since it contains the exact command and workspace/primary ids that will need to be run when installing the agent collector.

Run the deployment script

1. From the Azure Sentinel navigation menu, click **Data connectors**. From the list of connectors, click the **Common Event Format (CEF)** tile, and then the **Open connector page** button on the lower right.
2. Under **1.2 Install the CEF collector on the Linux machine**, copy the link provided under **Run the following script to install and apply the CEF collector**, or from the text below:

   ```
   sudo wget https://raw.githubusercontent.com/Azure/Azure-
   Sentinel/master/DataConnectors/CEF/cef_installer.py&&sudo python cef_installer.py
   [WorkspaceID] [Workspace Primary Key]
   ```

3. While the script is running, check to make sure you don't get any error or warning messages.

Note

Using the same machine to forward both plain Syslog *and* CEF messages

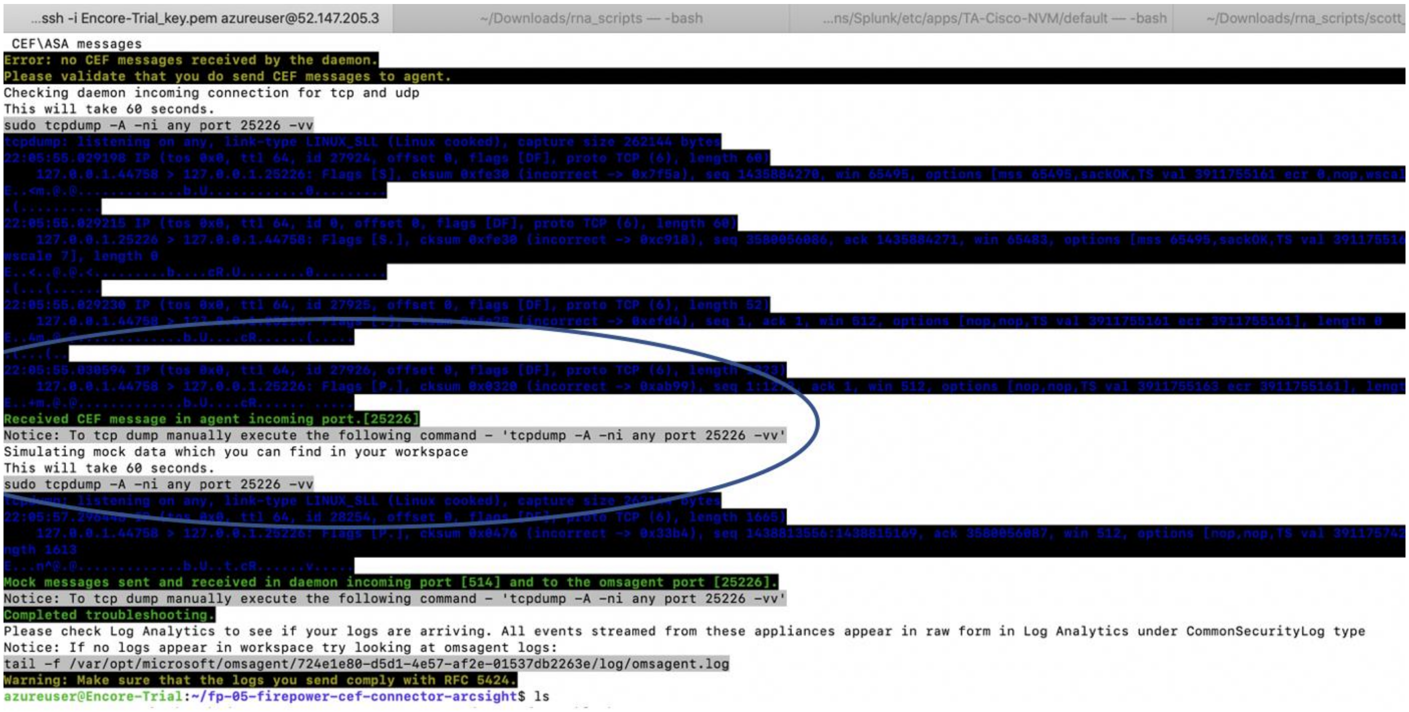
If you plan to use this log forwarder machine to forward Syslog messages as well as CEF, then in order to avoid the duplication of events to the Syslog and CommonSecurityLog tables:

1. On each source machine that sends logs to the forwarder in CEF format, you must edit the Syslog configuration file to remove the facilities that are being used to send CEF messages. This way, the facilities that are sent in CEF won't also be sent in Syslog. See Configure Syslog on Linux agent for detailed instructions on how to do this.
2. You must run the following command on those machines to disable the synchronization of the agent with the Syslog configuration in Azure Sentinel. This ensures that the configuration change you made in the previous step does not get overwritten.

   ```
   sudo su omsagent -c 'python /opt/microsoft/omsconfig/Scripts/OMS_MetaConfigHelper.py --
   disable'
   ```

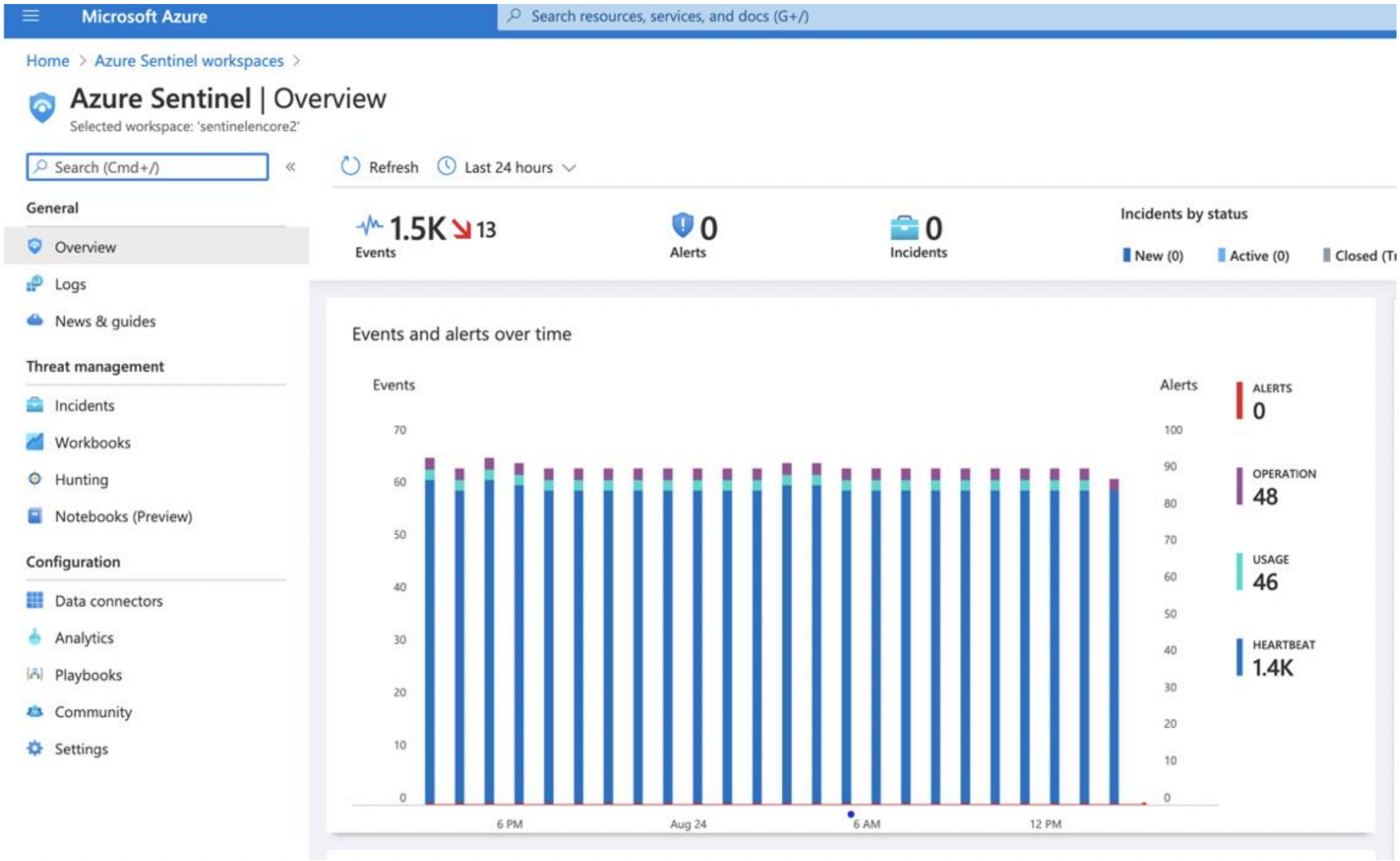
https://docs.microsoft.com/en-us/azure/sentinel/connect-cef-agent?tabs=rsyslog

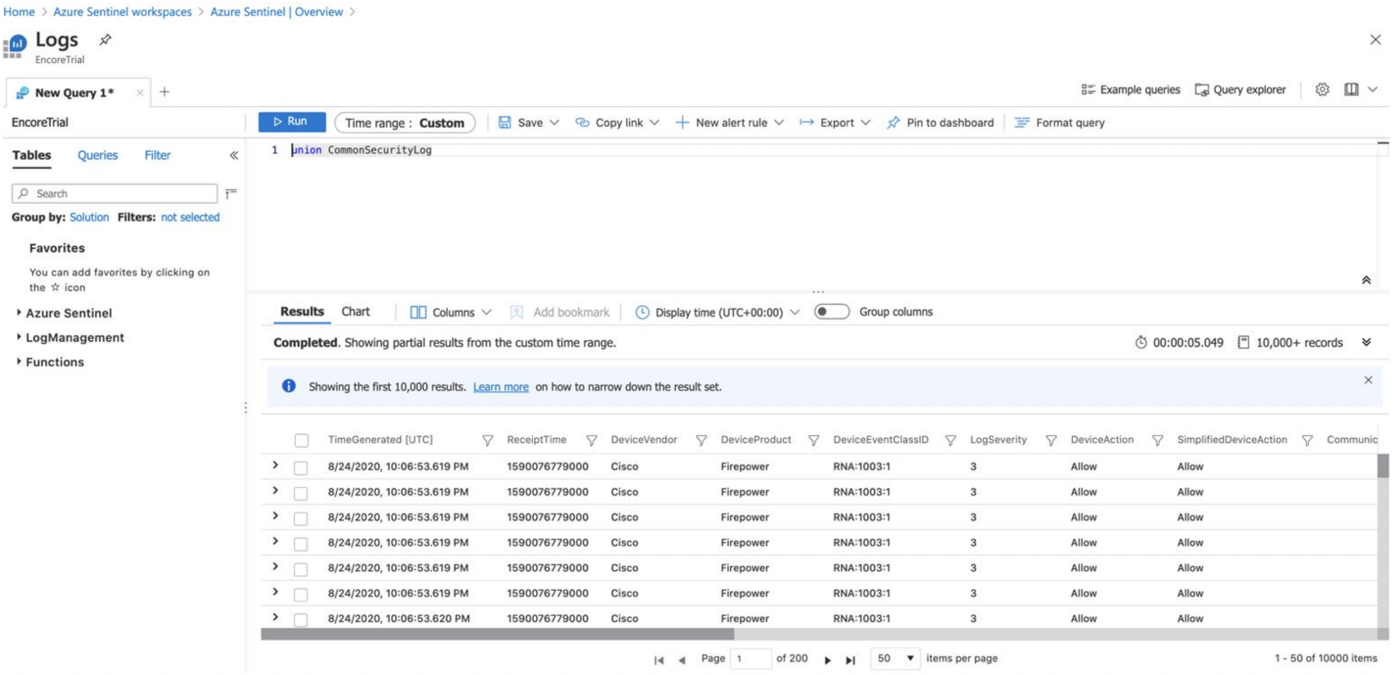
After running the validation script you should be able to see data coming into the Azure Sentinel Analytics screen


...ssh -i Encore-Trial_key.pem azureuser@52.147.205.3
~/Downloads/rna_scripts — -bash
...ns/Splunk/etc/apps/TA-Cisco-NVM/default — -bash
~/Downloads/rna_scripts/scott_

```
 CEF\ASA messages
Error: no CEF messages received by the daemon.
Please validate that you do send CEF messages to agent.
Checking daemon incoming connection for tcp and udp
This will take 60 seconds.
sudo tcpdump -A -ni any port 25226 -vv
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
22:05:55.029198 IP (tos 0x0, ttl 64, id 27924, offset 0, flags [DF], proto TCP (6), length 60)
    127.0.0.1.44758 > 127.0.0.1.25226: Flags [S], cksum 0xfe30 (incorrect -> 0x7f5a), seq 1435884270, win 65495, options [mss 65495,sackOK,TS val 3911755161 ecr 0,nop,wscal
E..<m.@.@.............b.U............0.........
.(..........
22:05:55.029215 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    127.0.0.1.25226 > 127.0.0.1.44758: Flags [S.], cksum 0xfe30 (incorrect -> 0xc918), seq 3580056086, ack 1435884271, win 65483, options [mss 65495,sackOK,TS val 391175516
wscale 7], length 0
E..<..@.@.<.........b....cR.U........0.........
.(...(......
22:05:55.029230 IP (tos 0x0, ttl 64, id 27925, offset 0, flags [DF], proto TCP (6), length 52)
    127.0.0.1.44758 > 127.0.0.1.25226: Flags [.], cksum 0xfe28 (incorrect -> 0xefd4), seq 1, ack 1, win 512, options [nop,nop,TS val 3911755161 ecr 3911755161], length 0
E..4m.@.@.............b.U....cR......(.....
.(...(..
22:05:55.030594 IP (tos 0x0, ttl 64, id 27926, offset 0, flags [DF], proto TCP (6), length 1223)
    127.0.0.1.44758 > 127.0.0.1.25226: Flags [P.], cksum 0x0320 (incorrect -> 0xab99), seq 1:1223, ack 1, win 512, options [nop,nop,TS val 3911755163 ecr 3911755161], lengt
E..+m.@.@.............b.U....cR...... .....
Received CEF message in agent incoming port.[25226]
Notice: To tcp dump manually execute the following command - 'tcpdump -A -ni any port 25226 -vv'
Simulating mock data which you can find in your workspace
This will take 60 seconds.
sudo tcpdump -A -ni any port 25226 -vv
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
22:05:57.296440 IP (tos 0x0, ttl 64, id 28254, offset 0, flags [DF], proto TCP (6), length 1665)
    127.0.0.1.44758 > 127.0.0.1.25226: Flags [P.], cksum 0x0476 (incorrect -> 0x33b4), seq 1438813556:1438815169, ack 3580056087, win 512, options [nop,nop,TS val 391175742
ngth 1613
E...n^@.@.............b.U..t.cR......v.....
Mock messages sent and received in daemon incoming port [514] and to the omsagent port [25226].
Notice: To tcp dump manually execute the following command - 'tcpdump -A -ni any port 25226 -vv'
Completed troubleshooting.
Please check Log Analytics to see if your logs are arriving. All events streamed from these appliances appear in raw form in Log Analytics under CommonSecurityLog type
Notice: If no logs appear in workspace try looking at omsagent logs:
tail -f /var/opt/microsoft/omsagent/724e1e80-d5d1-4e57-af2e-01537db2263e/log/omsagent.log
Warning: Make sure that the logs you send comply with RFC 5424.
azureuser@Encore-Trial:~/fp-05-firepower-cef-connector-arcsight$ ls
```

Note: Seeing the messing Received CEF message in agent (incoming port 25226) is an indicator that the validation and configuration of the agent was successful


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Azure Sentinel workspaces >
Azure Sentinel | Overview
Selected workspace: 'sentinelencore2'
Search (Cmd+/)
Refresh
Last 24 hours
General
Overview
Logs
News & guides
Threat management
Incidents
Workbooks
Hunting
Notebooks (Preview)
Configuration
Data connectors
Analytics
Playbooks
Community
Settings
1.5K 13
Events
0
Alerts
0
Incidents
Incidents by status
New (0)
Active (0)
Closed (T
Events and alerts over time
Events
Alerts
70
60
50
40
30
20
10
0
100
90
80
70
60
50
40
30
20
10
0
6 PM
Aug 24
6 AM
12 PM
ALERTS
0
OPERATION
48
USAGE
46
HEARTBEAT
1.4K



```
union CommonSecurityLog
```

Completed. Showing partial results from the custom time range. 00:00:05.049 10,000+ records

Showing the first 10,000 results. Learn more on how to narrow down the result set.

TimeGenerated [UTC]	ReceiptTime	DeviceVendor	DeviceProduct	DeviceEventClassID	LogSeverity	DeviceAction	SimplifiedDeviceAction	Communic
8/24/2020, 10:06:53.619 PM	1590076779000	Cisco	Firepower	RNA:1003:1	3	Allow	Allow	
8/24/2020, 10:06:53.619 PM	1590076779000	Cisco	Firepower	RNA:1003:1	3	Allow	Allow	
8/24/2020, 10:06:53.619 PM	1590076779000	Cisco	Firepower	RNA:1003:1	3	Allow	Allow	
8/24/2020, 10:06:53.619 PM	1590076779000	Cisco	Firepower	RNA:1003:1	3	Allow	Allow	
8/24/2020, 10:06:53.619 PM	1590076779000	Cisco	Firepower	RNA:1003:1	3	Allow	Allow	
8/24/2020, 10:06:53.619 PM	1590076779000	Cisco	Firepower	RNA:1003:1	3	Allow	Allow	
8/24/2020, 10:06:53.620 PM	1590076779000	Cisco	Firepower	RNA:1003:1	3	Allow	Allow	

7 Troubleshooting and questions

7.1 Error messages

As far as possible, eNcore has been engineered to provide meaningful error messages. Below is an example error message.

Figure 13: Example Error Message

The eStreamer service has closed the connection. There are a number of possible causes which may show above in the error log.

If you see no errors then this could be that
 * the server is shutting down
 * there has been a client authentication failure (please check that your outbound IP address matches that associated with your certificate - note that if your device is subject to NAT then the certificate IP must match the upstream NAT IP)
 * there is a problem with the server. If you are running FMC v6.0, you may need to install "Sourcefire 3D Defense Center S3 Hotfix AZ 6.1.0.3-1")

If you encounter errors that do not make sense or require further explanation, please contact support so that we can fix the problem and improve the error messages.

Microsoft Sentinel Agent install: If you encounter issues install the Microsoft agent on Azure then try reinstalling the OMS
 https://support.microsoft.com/en-us/help/4131455/how-to-reinstall-operations-management-suite-oms-agent-for-linux

7.2 Frequently Asked Questions

Can I output my data to a different server?

Yes. Currently eNcore only writes to the filesystem, but you could mount an NFS or SMB share and specify its path as above. This may impact performance.

Can I run more than one instance?

Yes, using the CLI version. Although currently the encore.sh shell script only supports one instance. The underlying Python program prefixes temporary files (e.g., metadata, certificates, bookmarks) with the host and port. You will also need to update the outputter locations (e.g., [Splunk] ... directory = splunk) in order to avoid data collision. If you wish to run more than one instance we recommend you extract additional copies of eStreamer-eNcore and configure separately in order to avoid changing encore.sh.

Can I connect to more than one FMC?

Currently not within a single instance. However, you can configure multiple instances as above.

Can eNcore de-duplicate data to keep my SIEM costs lower?

Not today. It is on the roadmap.

Can I run two instances of eNcore in a HA pair?

Yes and no. It is technically possible to run two side-by-side, but they will be completely ignorant of each other and output double the data. It may be preferable to run them in a hot-stand-by configuration where the primary client's state and configuration data is regularly copied to the secondary client. The state and configuration data in question is estreamer.conf; x.x.x.x-port_bookmark.dat; x.x.x.x-port_cache.dat; x.x.x.x-port_pkcs.cert; x.x.x.x-port_pkcs.key; x.x.x.x-port_status.dat

Can I increase the logging granularity?

Yes, change logging.level in the conf file. Please note that while it is possible to increase this level to VERBOSE, the performance impact will be crippling. DEBUG may be useful but slow. We strongly recommend not going above INFO for standard production execution.

8 Cisco Support

Support is provided by Cisco TAC.

9 Appendix A:

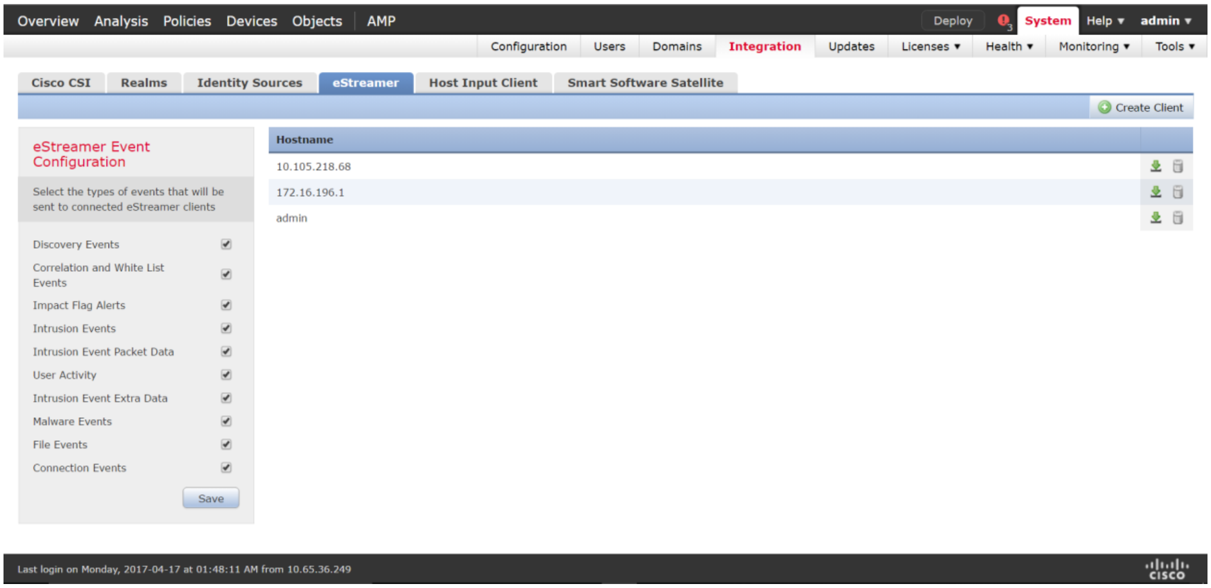
9.1 FMC eStreamer Certificate Creation

Steps to generate an eStreamer client certificate are as follows:

Navigate to the web interface of the FMC – https://fmc-ip-address and log in with your FMC credentials.

In the FMC 6.x GUI, navigate to System > Integration > eStreamer

Figure 14: FMC eStreamer Certificate Creation



eStreamer Event Configuration

Select the types of events that will be sent to connected eStreamer clients

- Discovery Events ☑
- Correlation and White List Events ☑
- Impact Flag Alerts ☑
- Intrusion Events ☑
- Intrusion Event Packet Data ☑
- User Activity ☑
- Intrusion Event Extra Data ☑
- Malware Events ☑
- File Events ☑
- Connection Events ☑

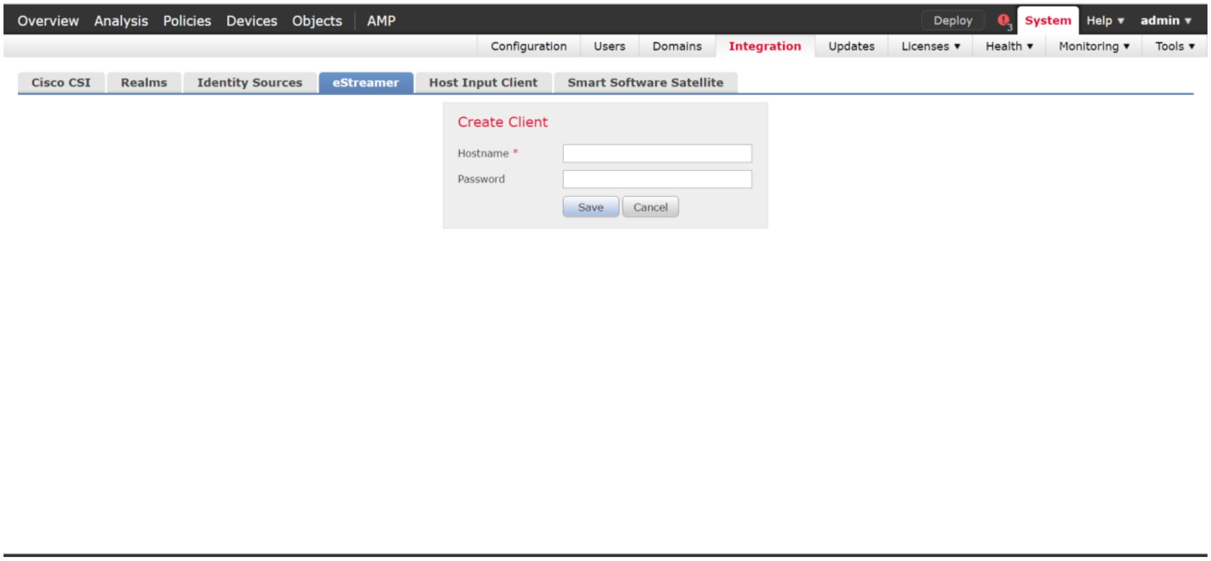
Hostname
10.105.218.68
172.16.196.1
admin

Last login on Monday, 2017-04-17 at 01:48:11 AM from 10.65.36.249

Click Create Client. Provide the Hostname and password.

Note: This should be the IP of the client, which will be collecting the event data from the FMC. This password will be required when you first execute eStreamer eNcore.
Please note that the IP address you enter here must be the IP address of the eStreamer-eNcore client *from the perspective of the FMC*. In other words, if the client is behind a NAT device, then the IP address must be that of the upstream NAT interface.

Figure 15: Create Client Hostname and Password Screen



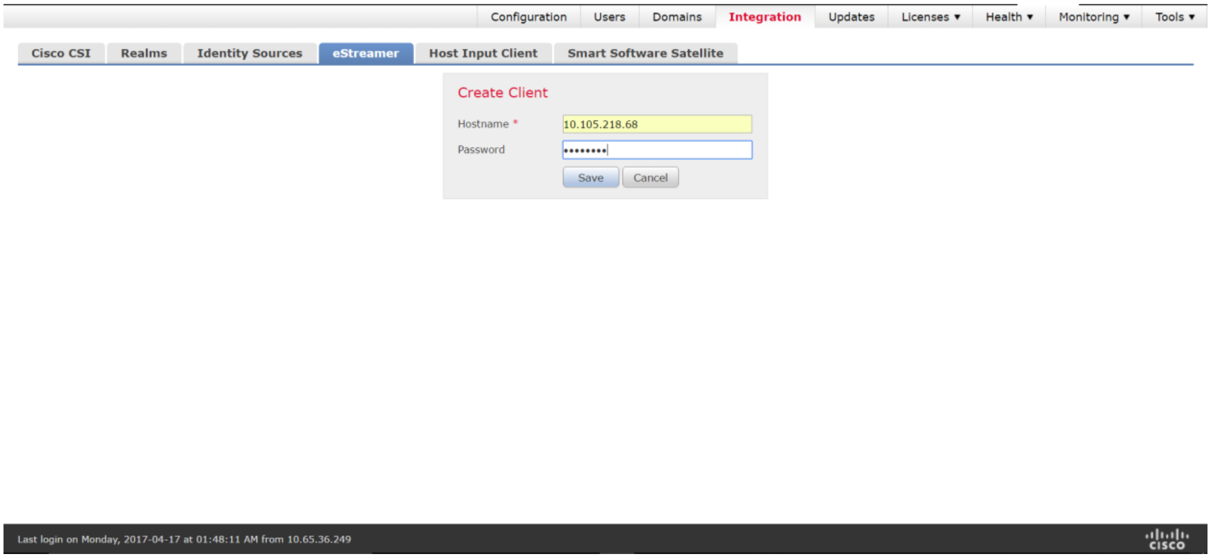
Create Client

Hostname *

Password

Click Save.

Figure 16: Create Client Save Screen



Create Client

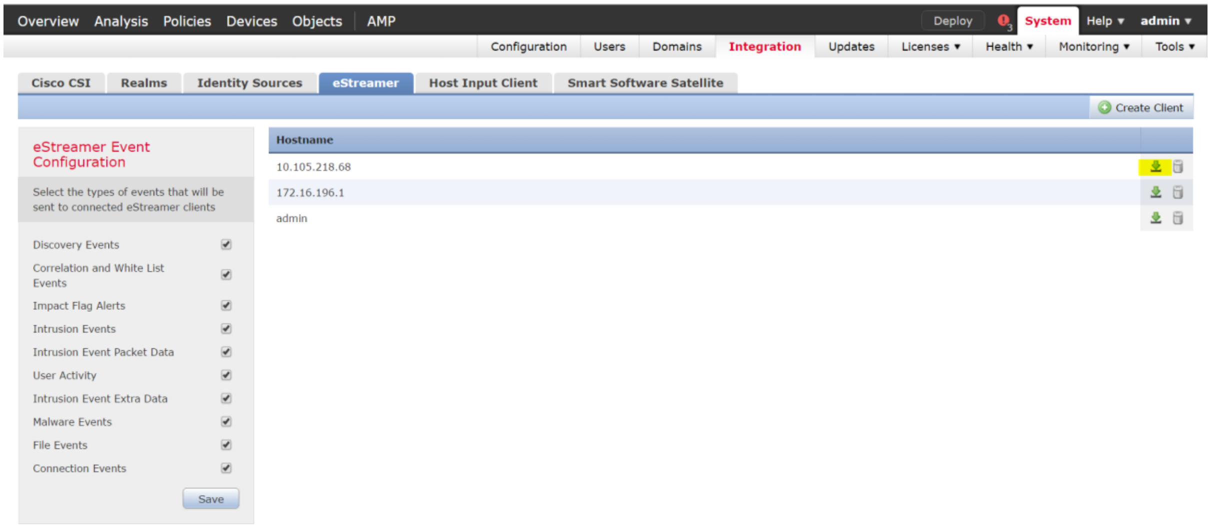
Hostname * 10.105.218.68

Password ••••••••

Last login on Monday, 2017-04-17 at 01:48:11 AM from 10.65.36.249

Download the pkcs12 file.

Figure 17: Download Screen



eStreamer Event Configuration

Select the types of events that will be sent to connected eStreamer clients

Discovery Events	☑
Correlation and White List Events	☑
Impact Flag Alerts	☑
Intrusion Events	☑
Intrusion Event Packet Data	☑
User Activity	☑
Intrusion Event Extra Data	☑
Malware Events	☑
File Events	☑
Connection Events	☑

Hostname
10.105.218.68
172.16.196.1
admin

Copy the pkcs12 file to the desired location in the target device. By default, eStreamer-eNcore will look for /path/eStreamer_eNcore/client.pkcs12. If you wish to use a different filename, then you must edit the estreamer.conf file.

9.2 Example Configuration File

Figure 18: Example Configuration File

```
{
  "connectTimeout": 10,
  "responseTimeout": 10,

  "@startComment": "0 for genesis, 1 for now, 2 for bookmark",
  "start": 2,


  "monitor": {
    "period": 120,
    "velocity": false,
    "bookmark": false,
    "subscribed": true,
    "handled": true
  },


  "logging": {
    "@comment": "Levels include FATAL, ERROR, WARNING, INFO, DEBUG, VERBOSE and TRACE",
    "level": "INFO",
    "format": "%(asctime)s %(name)-12s %(levelname)-8s %(message)s",
    "stdOut": true,
        "filepath": "estreamer.log"
    },


    "@queueComment": [
        "Maximum number of messages buffered before throttling takes place. The more powerful",
        "your CPU and more RAM you have, the larger this number can be. It's essentially a",
        "buffer size. Beyond a certain size you won't see any performance gain and it will",
        "just take longer to stop"
    ],

    "maxQueueSize": 100,

    "subscription": {
        "servers": [
            {
                "host": "1.2.3.4",
                "port": 8302,
                "pkcs12Filepath": "client.pkcs12",
                "@comment": "Valid values are 1.0 and 1.2",
                "tlsVersion": 1.2
            }
        ],


        "records": {
            "@comment": [
                "Just because we subscribe doesn't mean the server is sending. Nor does it mean",
                "we are writing the records either. See handler.records[]"
            ],
            "packetData": true,
            "extended": true,
            "metadata": true,
            "eventExtraData": true,
            "impactEventAlerts": true,
            "intrusion": true,
            "archiveTimestamps": true
        }
    },


    "handler": {
        "records": {
            "core": true,
            "metadata": true,
            "flows": true,
            "packets": true,
            "intrusion": true,
            "rua": true,
            "rna": true,
      "@includeComment": "These records will be included regardless of above",
      "include": [],


      "@excludeComment": [
        "These records will be excluded regardless of above (overrides 'include')",
        "e.g. to exclude flow and IPS events use [ 71, 400 ]"
      ],
      "exclude": []
    },


    "@comment": "If you disable all outputters it behaves as a sink",
    "outputters": [


      {
        "name": "CEF",
        "adapter": "cef",
        "enabled": true,
        "stream": {
          "uri": "udp://10.0.1.2:514",
        }
      },

      {
        "name": "CEFfile",
        "adapter": "cef",
        "enabled": true,
        "stream": {
          "uri": "relfile:///data/data.{0}.cef",
          "options": {
            "rotate": false,
            "maxLogs": 9999
          }
        }
      }

    ]
  }
}
```

Trademarks and Disclaimers

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE.

IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies are considered un-Controlled copies and the original on-line version should be referred to for latest version.

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)