SnappyFlow APM Installation

Getting Started Guide

Table of Contents

Table of Contents

Introduction 3

Prerequisites 3

Browser 3

Server 3

Hardware Spec for SnappyFlow APM server 3

Operating System Spec for SnappyFlow APM server 3

Networking Requirements 4

Packages Installed 4

Installation 8

Configure SnappyFlow APM Server 9

First time setup of SnappyFlow APM Server 9

Test your SnappyFlow APM Server initial configuration 9

Create Project 9

Create Application and monitor instance 10

Elasticsearch Cluster Installation 11

Hardware Spec for Elasticsearch server 11

Operating System Spec for Elasticsearch server 11

Networking Requirements 12

Server Setup 12

Install ElasticSearch rpm (Install Elasticsearch version 6.8.4) 13

Configure Elasticsearch 13

Configure user authentication for an ES Cluster 14

Add Elasticsearch cluster to an APM server 16

Elasticsearch Cluster Installation and configuration using Ansible 12

Introduction

SnappyFlow APM is an application monitoring platform. It supports both VM based applications and container applications. It works well with both private and public clouds. It is available as an AMI for AWS. It can also be installed on standalone servers. This document provides instructions to setup an APM server on a CentOS 7.2 server.

Prerequisites

Browser

This application has Web based user interface. Make sure you have latest version of one of the supported Browsers.

- Chrome version 35 (at minimum)
- Firefox version 26 (at minimum)

Server

Hardware Spec for SnappyFlow APM server

Below is the minimal hardware requirement for a demo version of APM server.

Server Specification	Value
vCPUs/Cores	4
RAM	8 GB
Disk Space	35 GB

For AWS, select t2.xlarge or higher.

Operating System Spec for SnappyFlow APM server

- Linux Distro: CentOS 7.2 or above
- It is suggested that the server should be a dedicated server with no other applications. Installation step upgrades/installs some of the libraries
- Access to root. Once logged into to server, sudo su should be allowed. All installations are done under root account
- Folder permissions. There should be permission to create below subfolders under these folder
    - Application is installed under /apps folder
    - Data folder is created under /data.
    - Logs are created under /var/log

Networking Requirements

- Egress Internet access from the instance
- Below Ingress ports should be opened
    - 22 (SSH), 80 (http), 443 (https), 9200 (Elasticsearch),
    - 5432 (postgresql)
- Server installation downloads and installs packages. It requires access to Internet or a proxy setup for corporate package repository. The commands used are standard yum install and pip install.
- Specific Internet Sites accessed.
    - http://yum.postgresql.org/
    - https://github.com
    - http://python.org/
    - https://packages.elastic.co
    - https://artifacts.elastic.co/

Packages Installed

Operating System (Cent OS) Packages/Software

SnappyFlow APM installs below packages

Package	**Version**
bzip2-devel	
db4-devel	
Elasticsearch	6.8.4
epel-release	7
expat-devel	
gcc	4.8.5
gdbm-devel	
git	
http://python.org/ftp/python/2.7.14/Python-2.7.14.tar.xz	2.7.14
http://yum.postgresql.org/9.6/redhat/rhel-6-x86_64/pgdg-redhat96-9.6-3.noarch.rpm	9.6.17
httpd	2.44
libffi-devel	3.0.13
libpcap-devel	
lsof	4.87
mod_ssl	2.4.6
mod_wsgi	3.4
ncurses-devel	
ntp	
openssl	1.0.2k
openssl-devel	1.0.2k
postgresql96	9.6.17
postgresql96-contrib	9.6.17

postgresql96-devel	9.6.17
postgresql96-libs	9.6.17
postgresql96-server	9.6.17
psmisc	22.20
python-devel	2.7.5
python-pip	20.0.2
readline-devel	
redis	3.2.12
sqlite-devel	
tk-devel	
Unzip	6.00
Wget	1.14
xz-devel	
zlib-devel	

Python PIP Packages

SnappyFlow APM installs below Python PIP Packages under system python.

PIP Package	Version
ansible	==2.6.11
awscli	>=1.17.14
ipaddr	==2.1.11
netaddr	==0.7.19
paramiko	>=2.4.1
requests	==2.10.0
scapy	==2.4.0
Setuptools	

Python PIP Packages installed in APM Python virtual environment

SnappyFlow APM installs below Python PIP Packages under APM virtual environment.

PIP Package	Version
gitpython	
python-keyczar	==0.716
boto3	==1.9.193
botocore	==1.12.193
django	==1.10.0
fabric	==1.14.0
django-cors-headers	>=2.1.0
scapy	==2.4.0
ansible	>=2.5.0,<2.7.0
requests	==2.10.0
celery	==4.1.0
kombu	==4.1.0
pexpect	==4.6.0
elasticsearch	==6.2.0
pyvmomi	==6.5.0
netaddr	==0.7.19
scp	==0.10.2
python-jenkins	==1.2.1
django-redis-cache	==1.7.1
django-extensions	==1.8.1
amqp	==2.2.1
paramiko	>=2.4.1
psycopg2	>=2.7.1,<2.8
django-rest-swagger	==2.1.2
djangorestframework	==3.6.2
oauth2client	==4.1.2
kafka	==1.3.3
google-api-python-client	==1.7.3
vine	==1.1.4
redis	==2.10.5
django-guardian	==1.4.9
ipaddr	==2.1.11
salt	>=2016.11.5
pytz	==2017.2
ruamel.yaml	==0.15.33
websocket-client	==0.39.0
django-jsonfield	==1.0.1
django-celery-beat	==1.0.1
django-celery-results	==1.0.1
mako	==1.0.6
slackclient	==1.3
PyGithub	==1.40
cryptography	>=2.2
ply	==3.10
billiard	==3.5.0.3

elasticsearch-dsl	>=6.0.0
kubernetes	==6.0.0

Installation

Follow below instruction to download and install SnappyFlow APM.

- SSH into your CentOS 7.2 server and execute below commands. Below instruction uses curl to download software. Install curl (command: yum install curl) or replace command with wget or similar command.

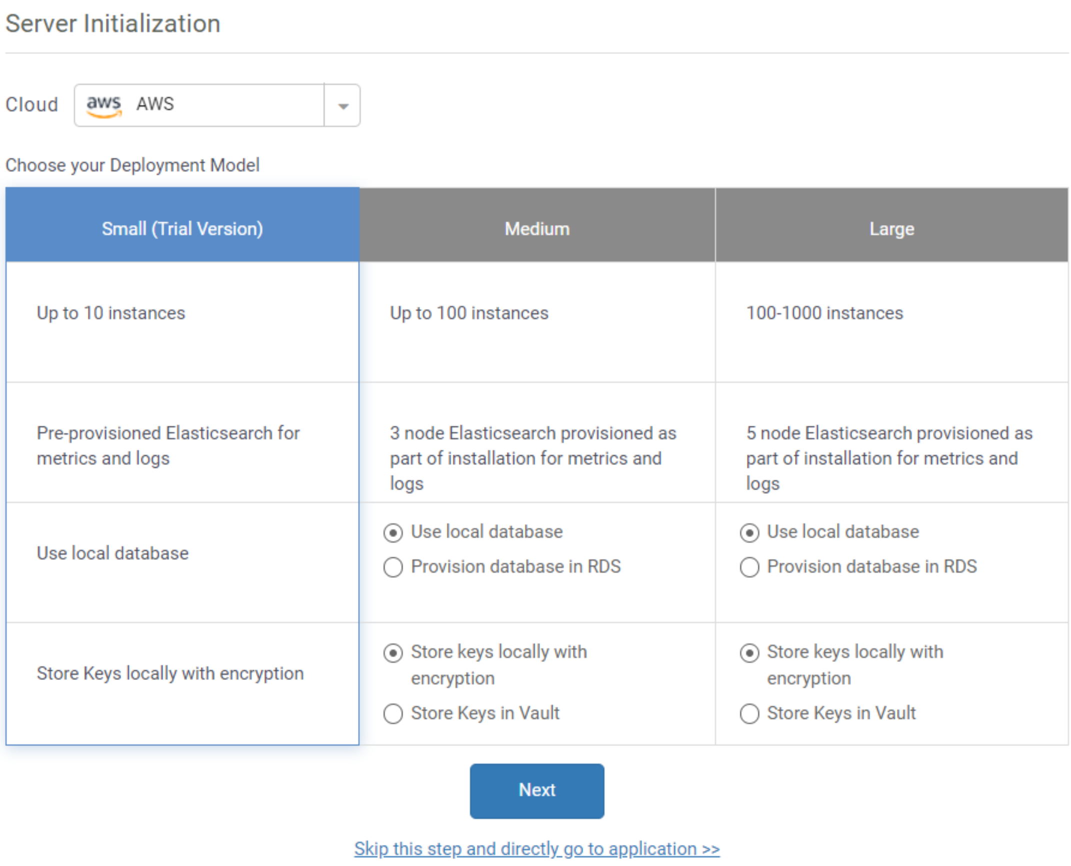
```
sudo su
cd /tmp
curl -u apmuser:apmpass
https://d2ll1pxudrx1z2.cloudfront.net/novartis/apm_integrated_build.tar.gz -o
apm_integrated_build.tar.gz
tar xzvf apm_integrated_build.tar.gz
chmod +x app_installer.sh
./app_installer.sh
```

- Installer extracts all components, downloads dependencies and installs apm server. Wait till you get a message Installation Success. It takes about 5-8 minutes to install components

Configure SnappyFlow APM Server

First time setup of SnappyFlow APM Server

- Open a Web browser.
- Connect to a newly deployed SnappyFlow APM server using URL http://<ip address>/
- Login using admin/admin
- Select Small (Trial Version) and follow the instructions to create first account which can be used to discover instance. You may skip this option, if you are not planning to discover instances from AWS.
- Below is a sample screen for server configuration



Server Initialization

Cloud: AWS

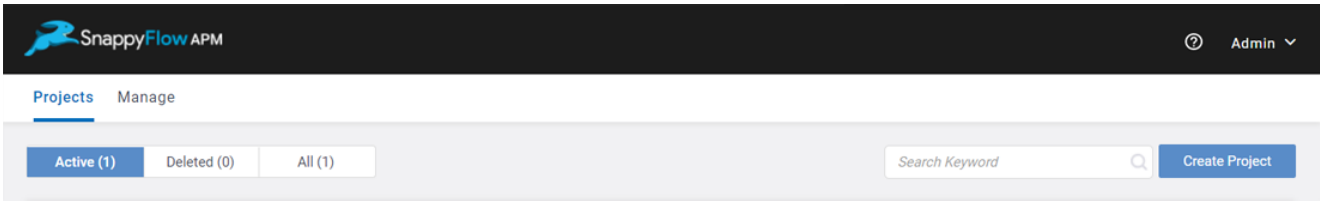
Choose your Deployment Model

Small (Trial Version)	Medium	Large
Up to 10 instances	Up to 100 instances	100-1000 instances
Pre-provisioned Elasticsearch for metrics and logs	3 node Elasticsearch provisioned as part of installation for metrics and logs	5 node Elasticsearch provisioned as part of installation for metrics and logs
Use local database	● Use local database ○ Provision database in RDS	● Use local database ○ Provision database in RDS
Store Keys locally with encryption	● Store keys locally with encryption ○ Store Keys in Vault	● Store keys locally with encryption ○ Store Keys in Vault

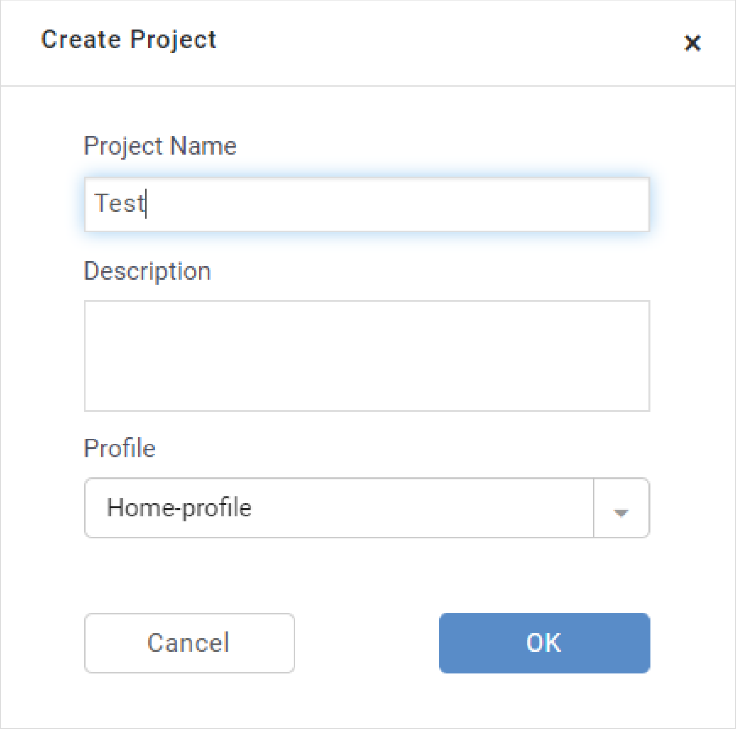
Test your SnappyFlow APM Server initial configuration

Create Project

- Go to Project Page
- Select Create project button



- Enter project name and click OK button.



Create Project

Project Name: Test

Description

Profile: Home-profile

Create Application and monitor instance

Create an application to monitor the SnappyFlow APM instance itself.

- In AWS console, for the newly launched instance, add tags projectName and appName
- In APM UI, Select Add application action for the newly created project
- Select option discover and add
- Enter AWS tags projectName and value
- Enter the tag key, appName for application tag.
- Add rule and click.
- Select the application to be added to the project.

Instance is discovered and monitoring starts for the instance.

Elasticsearch Cluster Installation

APM server install program installs a local single node Elasticsearch server instance on APM server. This is good for POC or experimentation. If you require more resilient Elasticsearch, use below steps to setup an ES cluster. You may vary the hardware spec based on the end system monitored.

Hardware Spec for Elasticsearch server

Below is the minimal hardware requirement for a demo version of APM server.

Number of instances: 3

Server Specification	Value
vCPUs/Cores	4
RAM	8 GB
Disk Space	100 GB (SSD)

If you are using AWS, you may select M5 node.

Operating System Spec for Elasticsearch server

- Linux Distro: Ubuntu 16
- It is suggested that the server should be a dedicated server with no other applications.
- Access to root. Once logged into to server, sudo su should be allowed. All installations are done under root account
- Java is installed
- Curl command is installed

Networking Requirements

- Egress Internet access from the instance
- Below Ingress ports should be opened
    - 22
- Ports 9200, and 9300 of all nodes should be accessible from the other nodes (configured in security groups in case of AWS)
- Server installation downloads and installs packages. It requires access to Internet or a proxy setup for corporate package repository. The commands used are standard yum install and pip install.
- Specific Internet Sites accessed.
    - https://packages.elastic.co
    - https://artifacts.elastic.co/

Elasticsearch Cluster Installation and configuration using Ansible

Ansible scripts simplify installation and configuration of Elasticsearch cluster. Follow below steps to install Elasticsearch.

Ansible Installation:

1. Identify a Linux VM from where you are going to run Ansible script. Make sure Ansible is installed on this node. Follow below steps to install Ansible

    ```
    sudo apt-add-repository ppa:ansible/ansible
    sudo apt-get update
    sudo apt-get install ansible -y
    ```

Install Elasticsearch

2. Note down the IP addresses of Linux VMs for ES cluster. Preferably use private IP addresses. Make sure these instances are reachable using SSH from Linux VM having Ansible.
3. On Ansible Linux VM, download the Ansible script.

    ```
    sudo su
    cd /tmp
    curl -u apmuser:apmpass
    https://d2ll1pxudrx1z2.cloudfront.net/novartis/ansible-es.tar.gz -o ansible-
    es.tar.gz
    tar -xzvf Ansible-es.tar.gz
    cd /ansible-es/2
    ```

4. Edit hosts file. Enter IP addresses of the Elasticsearch nodes.
5. Copy your SSH key to keys/provision.key and set Read permission

    ```
    chmod 400 provision.pem
    ```

6. Run Below command to install and configure Elasticsearch.

    ```
    chmod 777 ./script.sh
    ./script.sh
    ```

Installation and configuration of Elasticsearch takes about 10 minutes. Once installation completes, add cluster to APM server as mentioned in earlier chapter.

Manual Elasticsearch Server Setup (if Ansible is not preferred)

Create 3 Ubuntu instances and install elasticsearch as described below
Note down the IP addresses and designate these instances as master-1, master-2 and master-3

Example Node IP addresses used in below configuration. These may be different in your environment.

172.31.16.125
172.31.31.108
172.31.24.7

Install ElasticSearch rpm (Install Elasticsearch version 6.8.4)

Install and configure Elasticsearch on all 3 instances.

1. Install curl
   `sudo yum install curl`
2. sudo rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
3. sudo vi /etc/yum.repos.d/elasticsearch.repo

```
[Elasticsearch-6]
name=Elasticsearch repository for 6.x packages
baseurl=https://artifacts.elastic.co/packages/6.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

4. sudo yum install elasticsearch

Configure Elasticsearch

1. Edit config file in the path /etc/elasticsearch/elasticsearch.yml
   a. Decide name for a cluster. In below example, cluster name is systemapp-13308. Make sure cluster name is same in all instances.
   b. Make sure node.name is set to respective node name. Replace master-2 with respective name in below example.

```
bootstrap.memory_lock: false
cluster.name: systemapp-13308
discovery.zen.minimum_master_nodes: 2
discovery.zen.ping.unicast.hosts:
- 172.31.16.125
- 172.31.31.108
- 172.31.24.7
http.port: 9200
network.bind_host: 0.0.0.0
network.host: 0.0.0.0
node.data: true
node.master: true
node.name: master-2
transport.tcp.port: 9300

########### Paths ###############
```

```
# Path to directory containing configuration (this file and
logging.yml):

path.data: /var/lib/elasticsearch/master-2-master-2
path.logs: /var/log/elasticsearch/master-2-master-2

action.auto_create_index: true
```

2. Edit /etc/elasticsearch/master-2/jvm.options and set heap size. Leave default option for all other parameters. It is recommended to set heap to 50% of total RAM.

```
################################################################
## IMPORTANT: JVM heap size
################################################################
##
## You should always set the min and max JVM heap
## size to the same value. For example, to set
## the heap to 4 GB, set:
##
## -Xms4g
## -Xmx4g
##
## See https://www.elastic.co/guide/en/elasticsearch/reference/current/heap-
size.html
## for more information
##
################################################################

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space
-Xms4g
-Xmx4g
```

3. Configure Authentication for Elasticsearch
4. Restart elasticsearch service after making the above configuration changes on 3 nodes.

```
Service elasticsearch restart
```

5. Verify service status by using command,

```
Service elasticsearch status
```

Configure user authentication for an ES Cluster

1. **Enable xpack security on all nodes**

   In elasticsearch.yml –

   ```
   xpack.security.enabled: true
   ```

   Restart ES on all nodes

2. **Configure TLS on all nodes**

   Generate node certificates and enable TLS on each node by following the document: https://www.elastic.co/guide/en/elasticsearch/reference/6.8/configuring-tls.html#node-certificates

   Steps summary:

   - Generate certificate authority in master node and generate certificate for this node

- Scp ca file to all other nodes and generate certificates on each node
- Copy the certificates to the path <es-conf-path>/certs on each node
- Ensure that the certificate has 444 permissions
- Enable TLS in the config file

  In elasticsearch.yml –

  ```
  xpack.security.transport.ssl.enabled: true
  xpack.security.transport.ssl.verification_mode: none
  xpack.security.transport.ssl.keystore.path: certs/elastic-certificates.p12
  xpack.security.transport.ssl.truststore.path: certs/elastic-certificates.p12
  ```

3. **Generate passwords for built-in users using the following command:**

```
$ /usr/share/elasticsearch/ bin/elasticsearch-setup-passwords interactive
```

4. **Restart ES on all nodes**
5. **Create two roles in all nodes**

In roles.yml –

```
apmadmin:
  cluster:
    - all
  indices:
    -
      names:
        - "*"
      privileges:
        - all
monitorUser:
  cluster:
    - all
  indices:
    -
      names:
        - "*"
      privileges:
        - monitor
```

6. **Set anonymous user with apmadmin role for all nodes**

In elasticsearch.yml –

```
xpack:
  security:
    authc:
      realms:
        native1:
          type: native
          order: 0
      anonymous:
        username: anonymous_user
        roles: apmadmin
        authz_exception: false
```

7. **Create a user from the master node**

```
$ curl -X POST "localhost:9200/_xpack/security/user/apmuser " -H 'Content-Type: application/json' -d'
```

```
{
    "password": "apmpass",
     "roles”: ["apmadmin"],
     "full_name": "APM User"
}’
```

8. **Update ES config and set anonymous_user to use monitorRole on all nodes**

   In elasticsearch.yml –

```
xpack:
  security:
    authc:
      realms:
        native1:
          type: native
          order: 0
      anonymous:
        username: anonymous_user
        roles: monitorRole
        authz_exception: false
```

9. **All indices create, search and delete queries should be accessible only with the credentials:**

   Username - apmuser

   Password – apmpass

Add Elasticsearch cluster to an APM server

    - Login as admin.
    - Go to Manage -> ES Clsuters
    - Add ES-Cluster
- Create a new profile under Manage->Profile and use this newly added elastic search cluster. This new profile can be used to create projects.