AppSentinels.ai
Application Security Re-invented

AppSentinels API Security Platform

Backup & restore

Contents

1. Introduction..... 4
2. Method 1: Platform directory restore to New Instance:..... 4
3. Method 2: Snapshot and Restore to New Instance 9

Revision	Date Modified	Author	Comments
1.0	28-Oct-24		Manjunatha

1. Introduction

The document outlines two main methods:

1. Platform directory restore to New Instance:
   - Takes a backup of the as_platform directory
   - Involves creating a new user with sudo access (optional)
   - Requires checking and configuring proxy settings
   - Installation of Docker and docker-compose
   - Pulling AppSentinels images using a login token
   - Setting up necessary user groups and permissions
   - Configuring the hostname and IP addresses
   - Bringing up AppSentinels services using docker-compose
2. Snapshot and Restore to New Instance (AWS EC2 focused):
   - Creating a volume snapshot from the existing instance
   - Creating a new volume from the snapshot
   - Setting up a new EC2 instance
   - Detaching the original volume and attaching the snapshot-created volume
   - Updating IP configurations in docker-compose
   - Starting the services on the new instance

Please note that

- Both the above methods preserve existing data and configurations
- Users can log in with the same credentials after restoration
- Requires careful attention to IP address updates in configurations
- Involves Docker container management
- Includes specific steps for different operating systems (Ubuntu vs RHEL)

2. Method 1: Platform directory restore to New Instance:

1. On Running Platform, please take a back up the as_platform directory as a

On a New Instance follow the below steps

2. Create user with sudo access (OPTIONAL, recommended if allowed to create separate user or use existing user with sudo priviliges)

Ignore this step if the user is already created.

- sudo useradd -m -s /bin/bash appsentinels

- sudo passwd appsentinels

- sudo usermod -aG sudo appsentinels (if Ubuntu)

sudo usermod -aG wheel appsentinels (if REDHAT)

- su - appsentinels (change to appsentinels user)

```
[ec2-user@ip-10-101-3-250 ~]$ sudo useradd  -m -s /bin/bash appsentinels
[ec2-user@ip-10-101-3-250 ~]$ sudo passwd appsentinels
Changing password for user appsentinels.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-101-3-250 ~]$ sudo usermod -aG wheel appsentinels
[ec2-user@ip-10-101-3-250 ~]$ su - appsentinels
Password:
[appsentinels@ip-10-101-3-250 ~]$
```

3. Check the proxy settings

- Check the proxy configuration
- if proxy enabled, it should be disabled or excluded for localhost & HOST IP
- Docker environment should not be using proxy

4. Download the AppSentinels Platform installation files to home directory of appsentinels

wget appsentinels-platform-installation-XX.XX.RX.tar.gz

5. Install docker & docker-compose

- cd pre-req
- bash docker-install.sh
- logout & login. Change user back to appsentinels

6. Pull appsentinels images

- cd pre-req
- Please run below command to pull the docker images. Please Ask Appsentinels Team for the logintoken

bash appsentinelsimages.sh <logintoken>


```
[appsentinels@ip-10-101-3-250 ~]$ cd installation/
[appsentinels@ip-10-101-3-250 installation]$ ls
certs  check_list.txt  config_files  deployment_src  install_saas  pre-req  readme.txt  utils
[appsentinels@ip-10-101-3-250 installation]$ cd pre-req/
[appsentinels@ip-10-101-3-250 pre-req]$ ls
appsentinelsimages.sh              docker-install-rhel.sh  docker-install-ubuntu.sh  other-utils-install-rhel.sh
configure_docker_to_use_proxy.txt  docker-install.sh       docker-rpm-download.txt   other-utils-install-ubuntu.sh
[appsentinels@ip-10-101-3-250 pre-req]$ ls -lrt
total 36
-rw-rw-r-- 1 appsentinels appsentinels  170 Oct 25 11:04 other-utils-install-ubuntu.sh
-rw-rw-r-- 1 appsentinels appsentinels  151 Oct 25 11:04 other-utils-install-rhel.sh
-rw-rw-r-- 1 appsentinels appsentinels 1301 Oct 25 11:04 docker-rpm-download.txt
-rw-rw-r-- 1 appsentinels appsentinels 2192 Oct 25 11:04 docker-install-ubuntu.sh
-rw-rw-r-- 1 appsentinels appsentinels 5164 Oct 25 11:04 docker-install.sh
-rw-rw-r-- 1 appsentinels appsentinels  708 Oct 25 11:04 docker-install-rhel.sh
-rw-rw-r-- 1 appsentinels appsentinels  631 Oct 25 11:04 configure_docker_to_use_proxy.txt
-rw-rw-r-- 1 appsentinels appsentinels 2444 Oct 25 11:04 appsentinelsimages.sh
[appsentinels@ip-10-101-3-250 pre-req]$ bash docker-install.sh
Docker not found. Installing Docker...

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for appsentinels:
```

```
[appsentinels@ip-10-101-3-250 installation]$ cd pre-req/
[appsentinels@ip-10-101-3-250 pre-req]$ bash appsentinelsimages.sh
```

7. **if RHEL or AL2 then run below commands**

sudo groupadd docker-sentinels

sudo groupadd data-sentinels

sudo groupadd data-analysis

sudo usermod -aG docker-sentinels $USER

sudo usermod -aG data-sentinels $USER

sudo usermod -aG data-analysis $USER

sudo usermod -aG adm $USER

Other OS run below commands

sudo addgroup docker-sentinels

sudo addgroup data-sentinels

sudo addgroup data-analysis

sudo adduser $USER docker-sentinels

sudo adduser $USER data-sentinels

sudo adduser $USER data-analysis

sudo adduser $USER adm

```
[appsentinels@ip-10-101-3-250 /]$ sudo groupadd docker-sentinels
[sudo] password for appsentinels:
[appsentinels@ip-10-101-3-250 /]$ sudo groupadd data-sentinels
[appsentinels@ip-10-101-3-250 /]$ sudo groupadd data-analysis
[appsentinels@ip-10-101-3-250 /]$ sudo usermod -aG docker-sentinels $USER
[appsentinels@ip-10-101-3-250 /]$ sudo usermod -aG data-sentinels $USER
[appsentinels@ip-10-101-3-250 /]$ sudo usermod -aG data-analysis $USER
[appsentinels@ip-10-101-3-250 /]$ sudo usermod -aG adm $USER
[appsentinels@ip-10-101-3-250 /]$
```

8. Logout & login

9. untar the as_platform back up taken from prepious instance and move it to BASE_DIR path

```
[appsentinels@ip-10-101-3-250 /]$ sudo tar -xf as_platform.tar.gz
```

10. Navigate to /as_platform/deployment folder

11. The hostname should be DNS resolvable, if not add entry to /etc/hosts in the server

12. Edit the docker-compose.yaml file and change the IP which is referenced under extra_hosts in entire file to new IP

```
    environment:
      - 'DATABASE_URI=postgresql://kong:kong@postgres:5432/policy'
      - policy_component_url=https://al2.appsentinels.ai/
    expose:
      - 5002
    restart: on-failure
    networks:
      ingestion:
        aliases:
          - scheduler
    extra_hosts:
      - "al2.appsentinels.ai:10.101.3.250"
    secrets:
      - devops_connection
    logging:
      driver: syslog
      options:
        tag: docker-scheduler
  userinterface:
    image: appsentinelsai/appsentinels-ui:24.09.R1
    expose:
      - 80
    restart: on-failure
    networks:
      ingestion:
        aliases:
          - userinterface
    environment:
      - AUTH_KEY=keycloak
      - DOMAIN=https://al2.appsentinels.ai/auth
      - CLIENTID=frontend
      - REALM=Appsentinels
      - SERVERURL=https://al2.appsentinels.ai
    extra_hosts:
      - "al2.appsentinels.ai:10.101.3.54"
    deploy:
      restart_policy:
        condition: on-failure
      resources:
        limits:
                                                                769,30        55%
```

13. if sensor is deployed in the same host, then change the REMOTE_CONTROLLER_SERVER_NAME to new IP, otherwise skip this step

```
# docker-compose -f docker-compose-sniffer-sensor.yaml pull && docker-co
#
#installing docker-compose sudo curl -L "https://github.com/docker/compo
er-compose
#https://docs.docker.com/compose/environment-variables/#substitute-envir

version: "3.3"
services:
  sniffer:
    container_name: sniffer-sensor
    restart: on-failure:5
    image: appsentinels/ng-controller:latest
    hostname: appsentinels-sniffer-sensor
    environment:
      - REMOTE_CONTROLLER_SERVER_NAME=10.101.3.250
      - REMOTE_CONTROLLER_SERVER_PORT=9006
      - ENVIRONMENT=scale
      - TAP_INTERFACE=default
      - TAP_FILTER=port-8000-and-tcp
      - RELAY_PROTOCOL=http
    network_mode: "host"
    logging:
      driver: local
      options:
        max-size: 10m
    volumes:
      - /var/crash:/var/crash
      - /var/log/appsentinels:/var/log/appsentinels
  http_service:
```

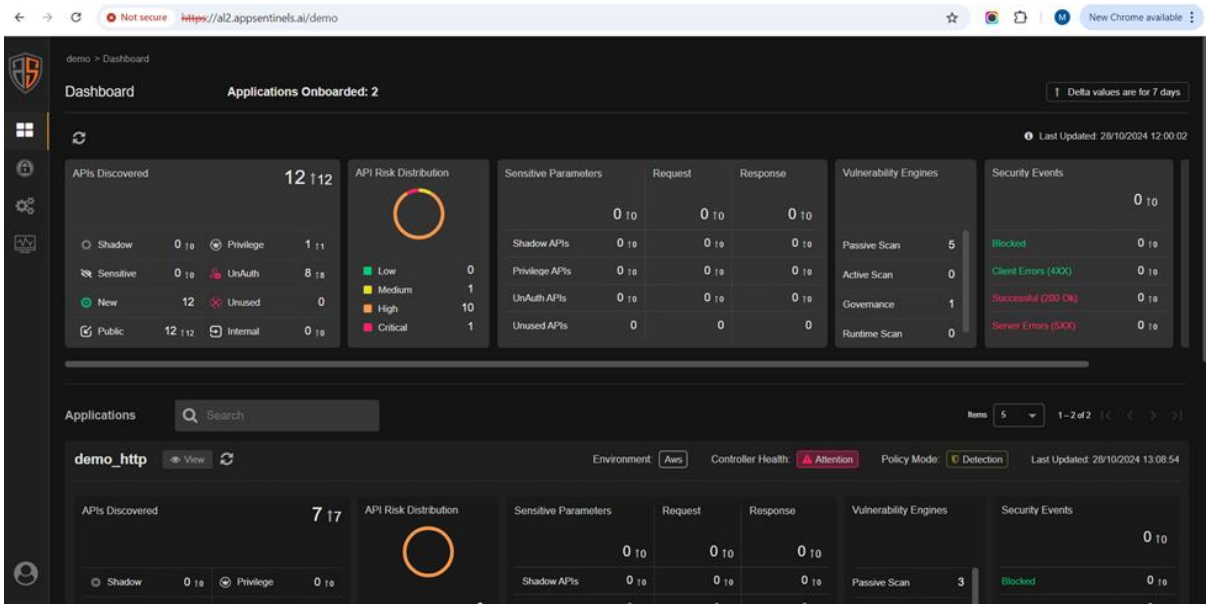
14. bring up the appsentinels services by running below command

docker-compose up –d

15. bring up the sensor if is deployed in the same instance otherwise skip this step

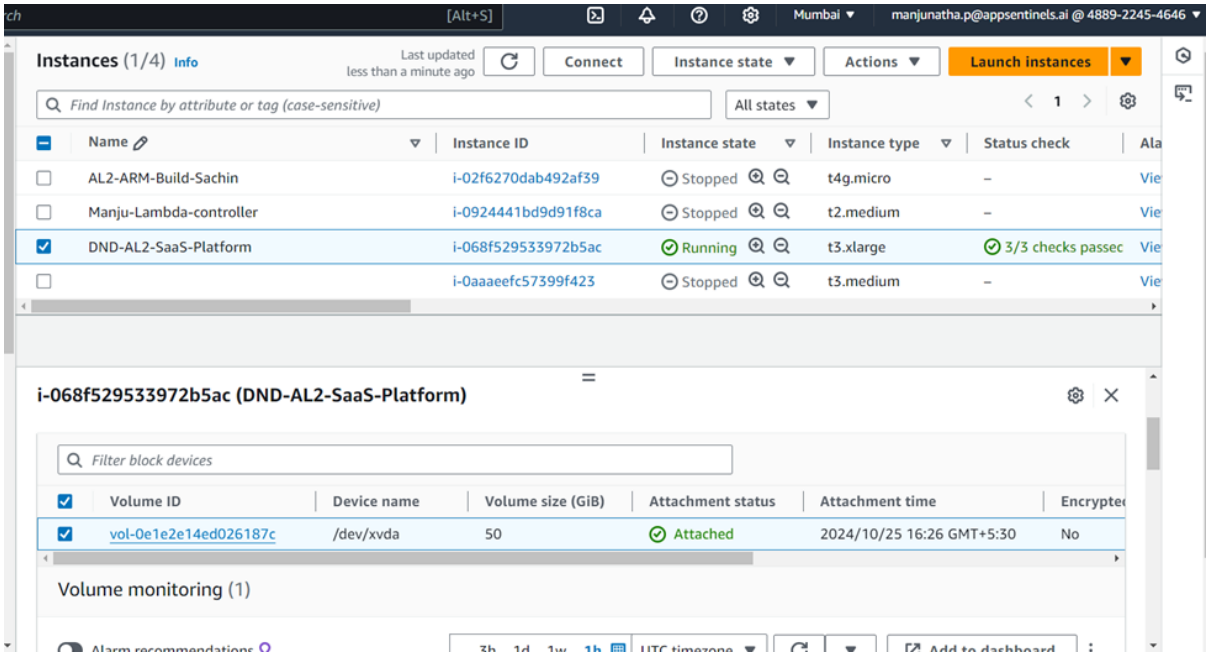
docker-compose –f <filename> up –d

16. Login to Appsentinels Platform with same credentials

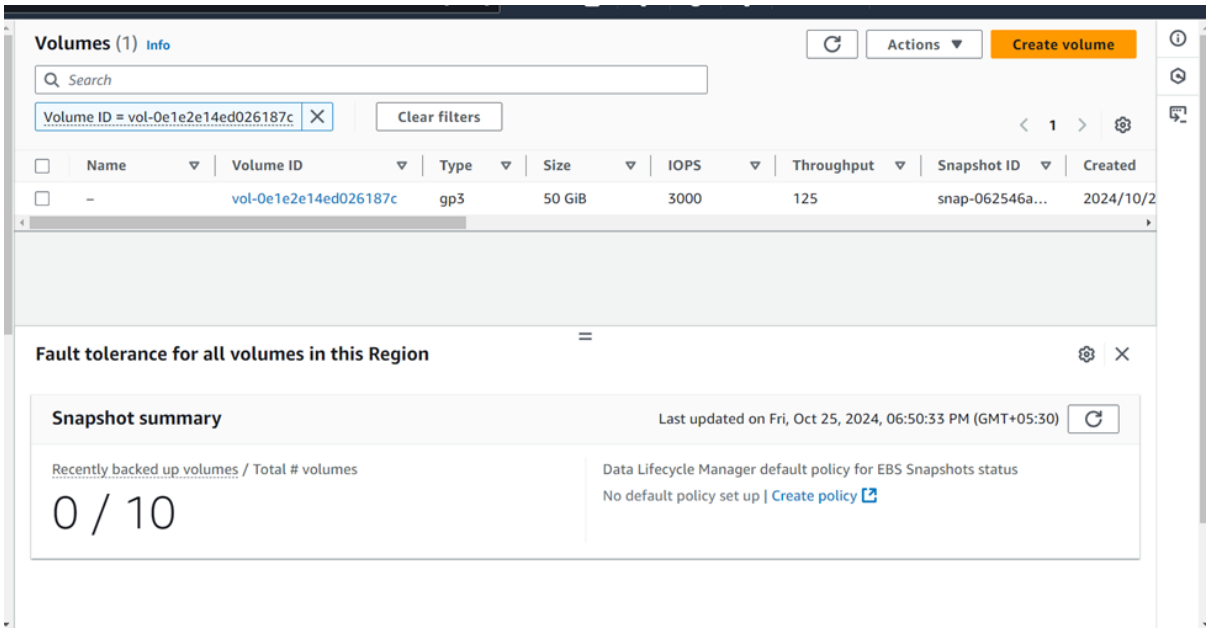


3. Method 2: Snapshot and Restore to New Instance

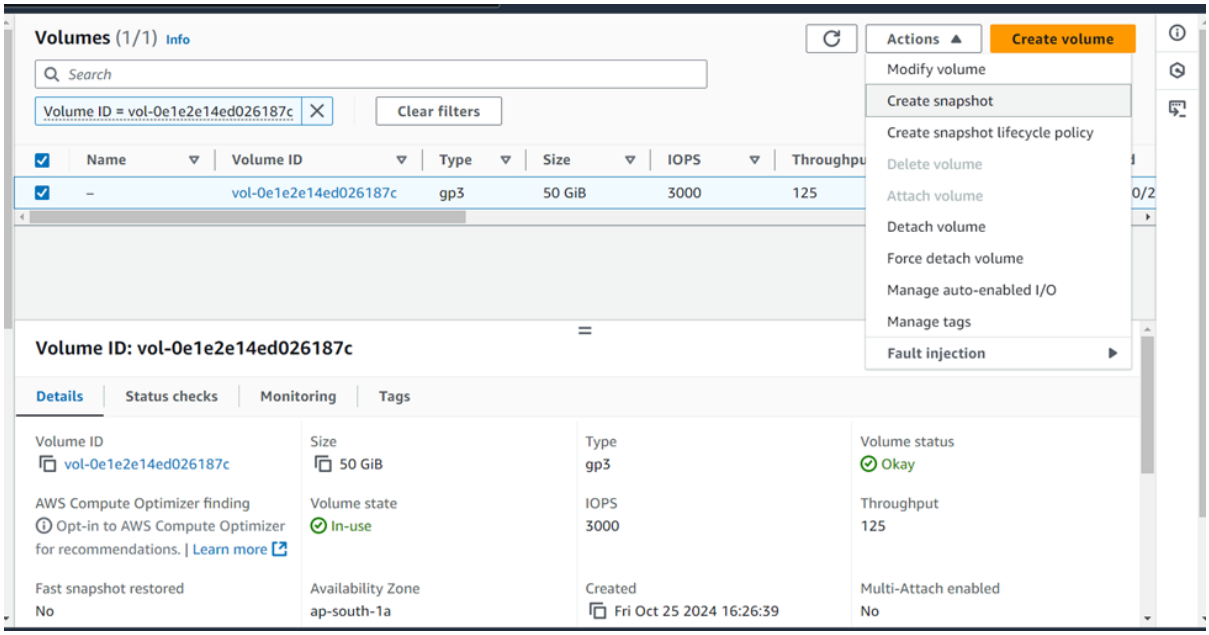
1. Select the Instance, Click on Storage, Click on the Volume



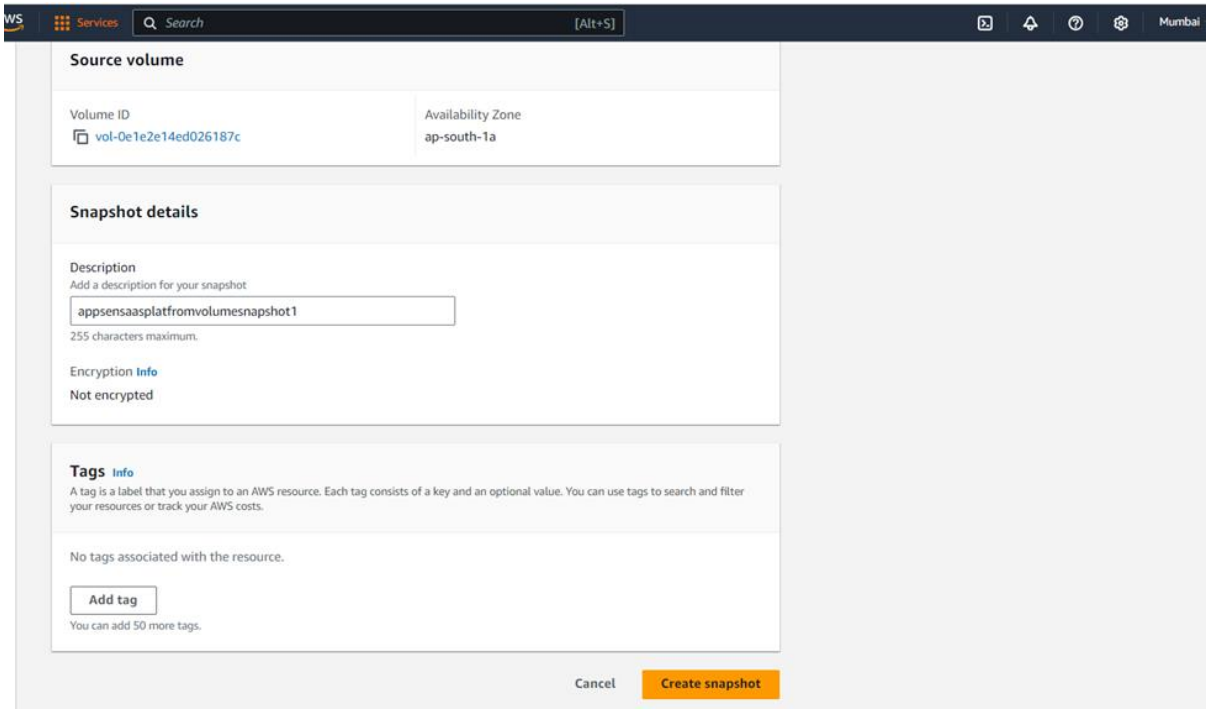
2. Volume is displayed



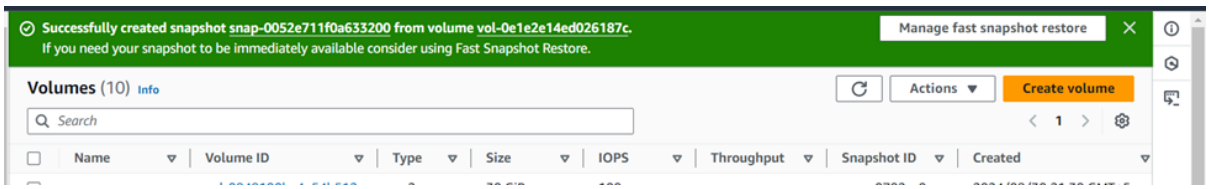
3. Select the Volume & Select Create Snapshot in Actions Dropdown



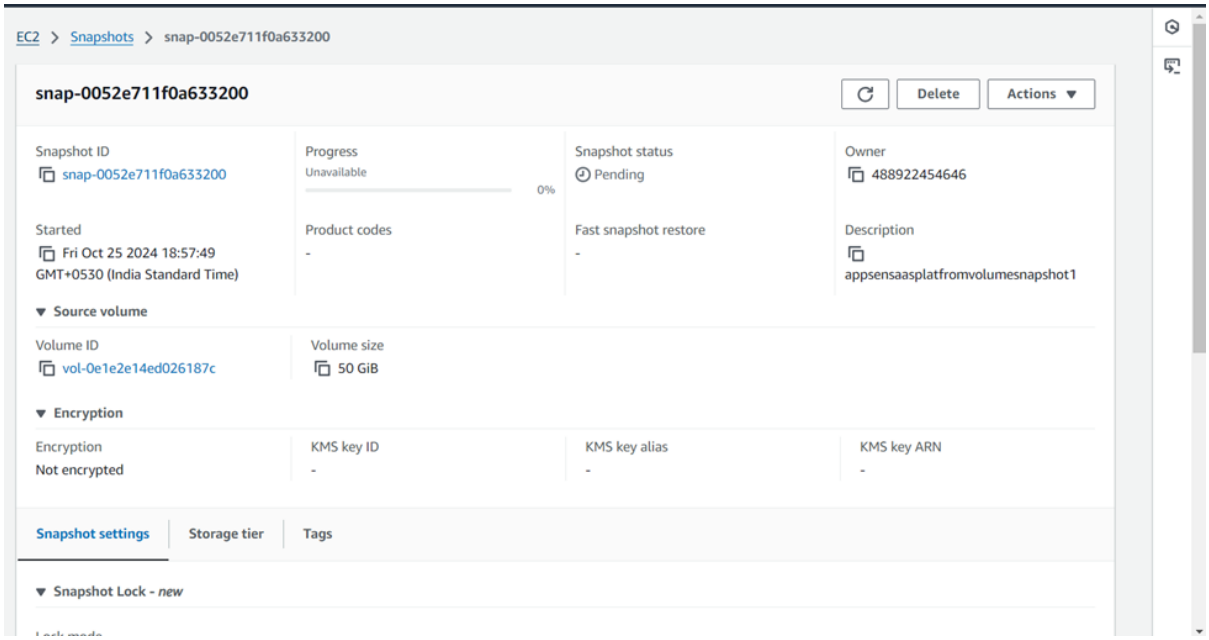
4. Provide the Details & click on Create Snapshot



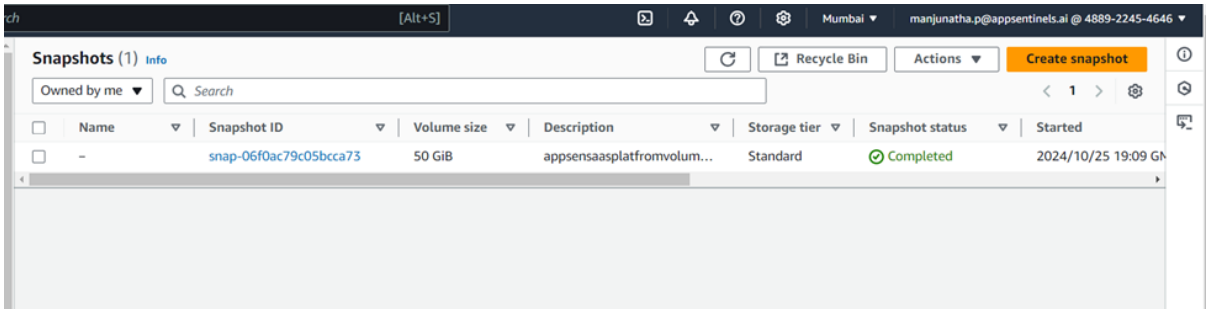
5. Snapshot will be created and click on the snapshot click



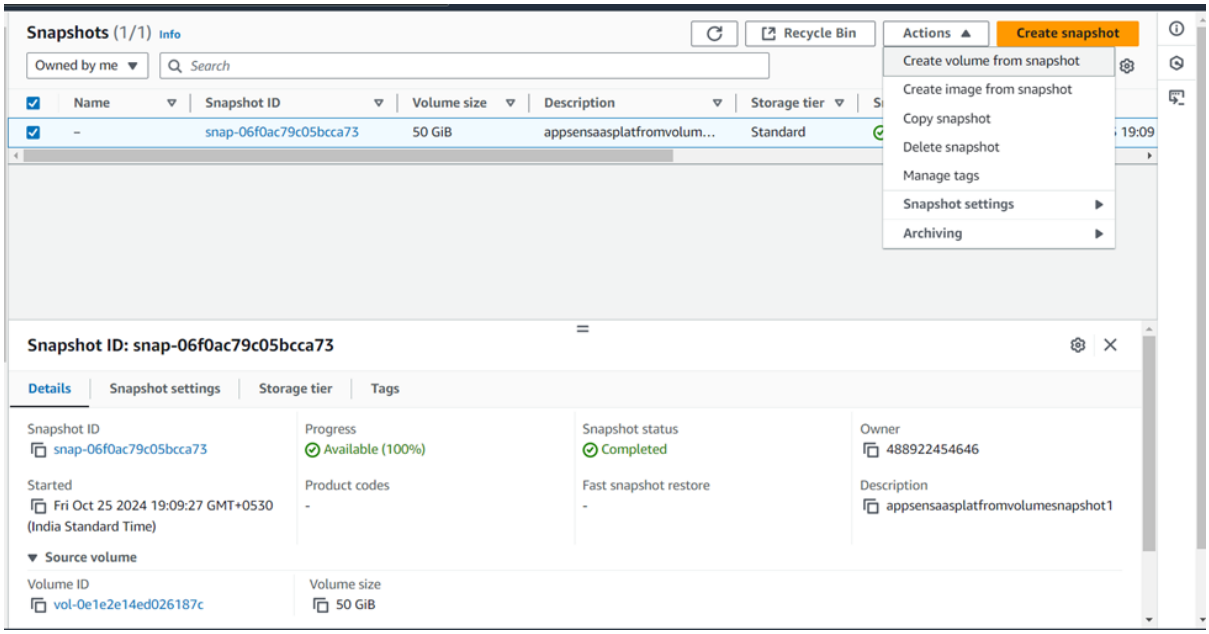
6. It will take some time to create snapshot



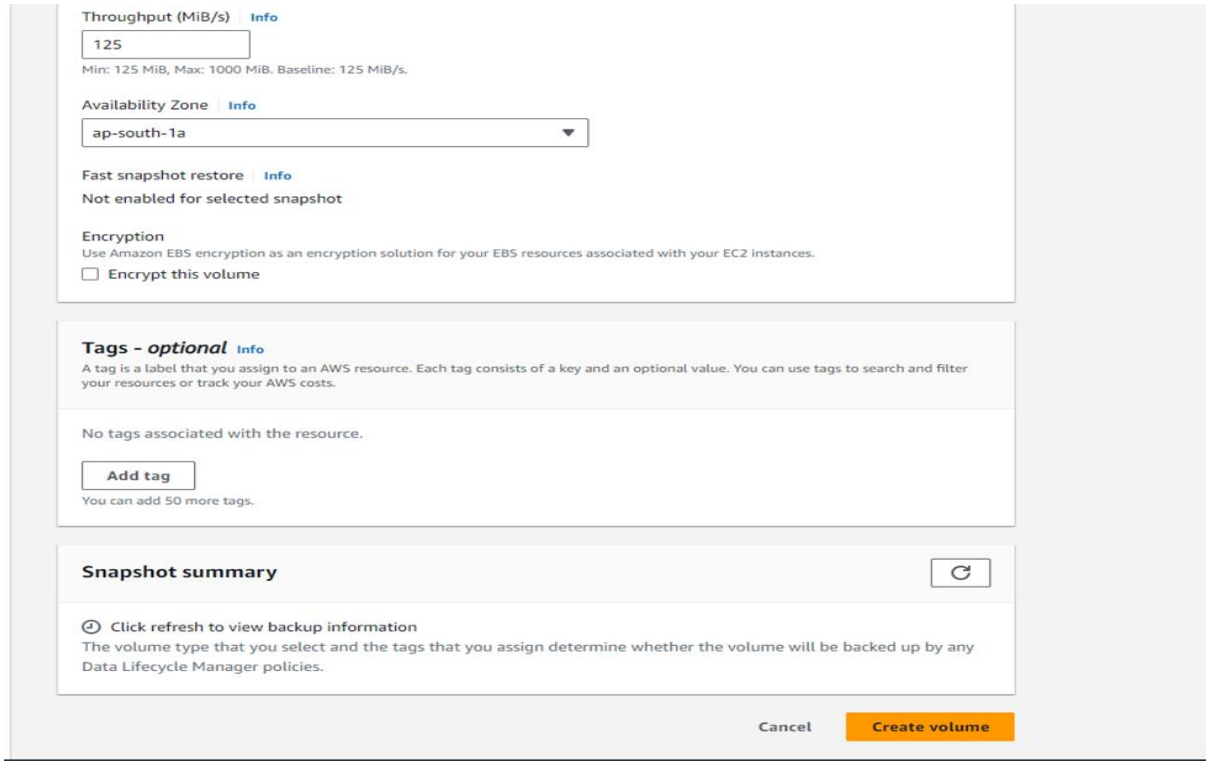
7. Once Snapshot is ready it will show status as Completed



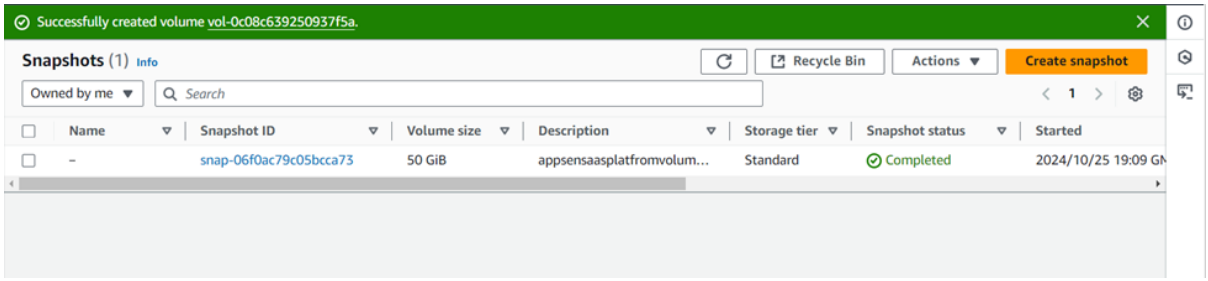
8. Select the snpshot & Click on Create Volume from snapshot in Actions dropdown



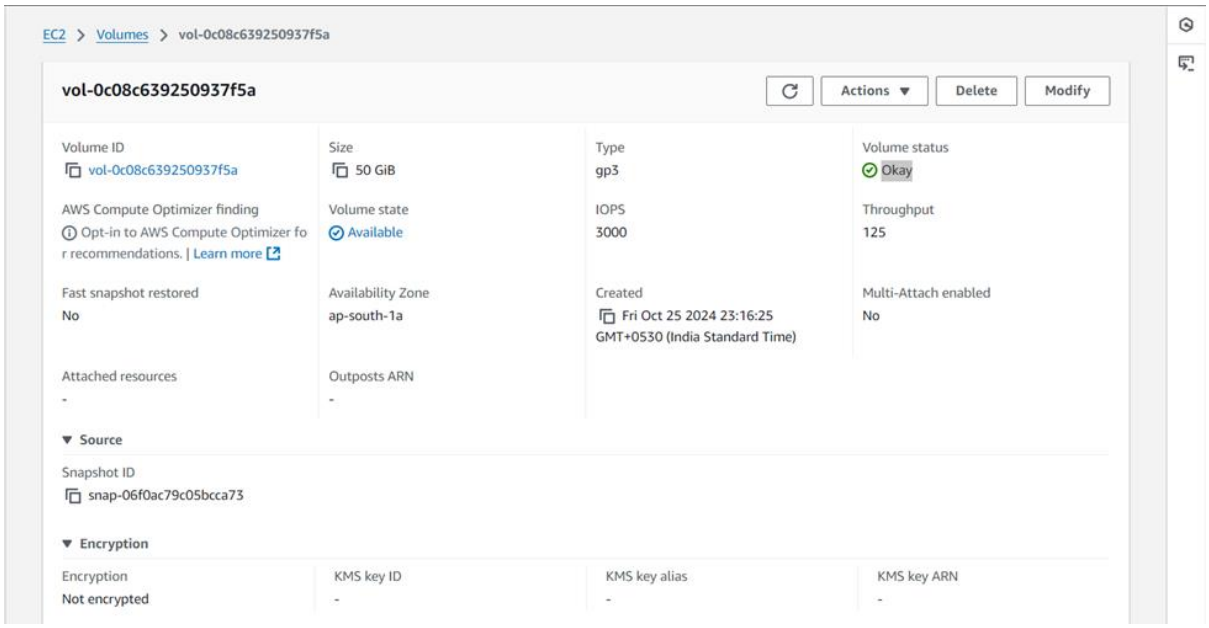
9. Provide the Details & click on Create Volume



10. It will show the create volumn and click on it

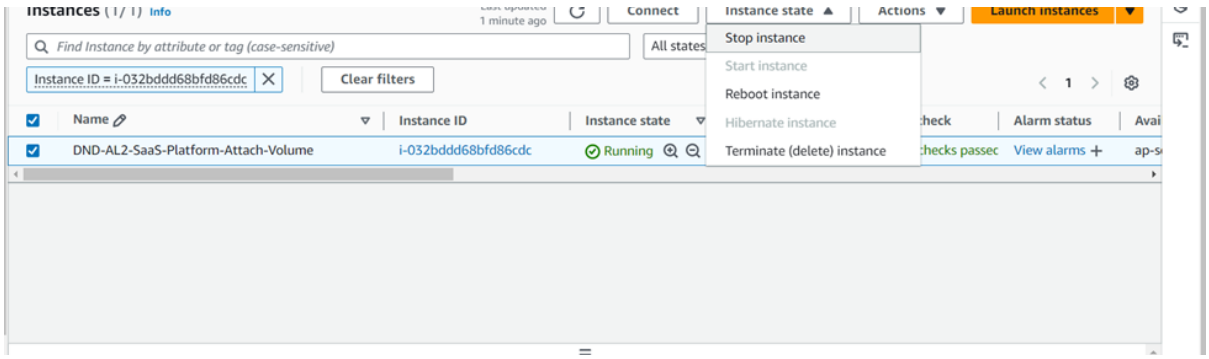


11. Wait for Volume status to be Okay

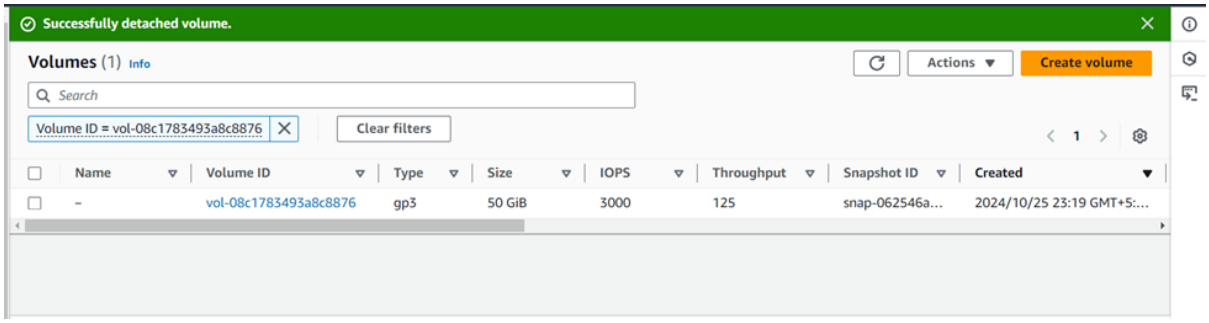


12. Stop the Previous running SaaS Platform Instance

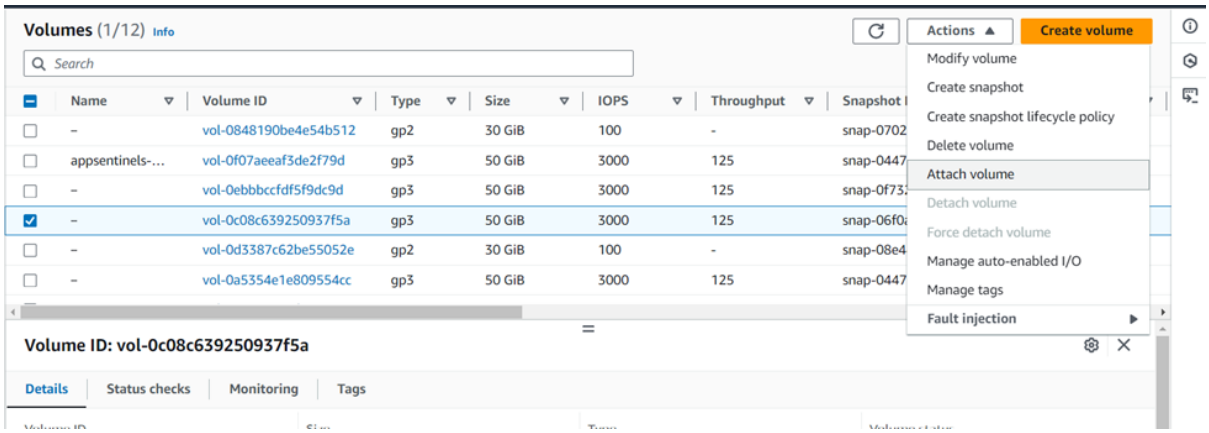
13. Create the New Ec2 Instance & Stop the instance



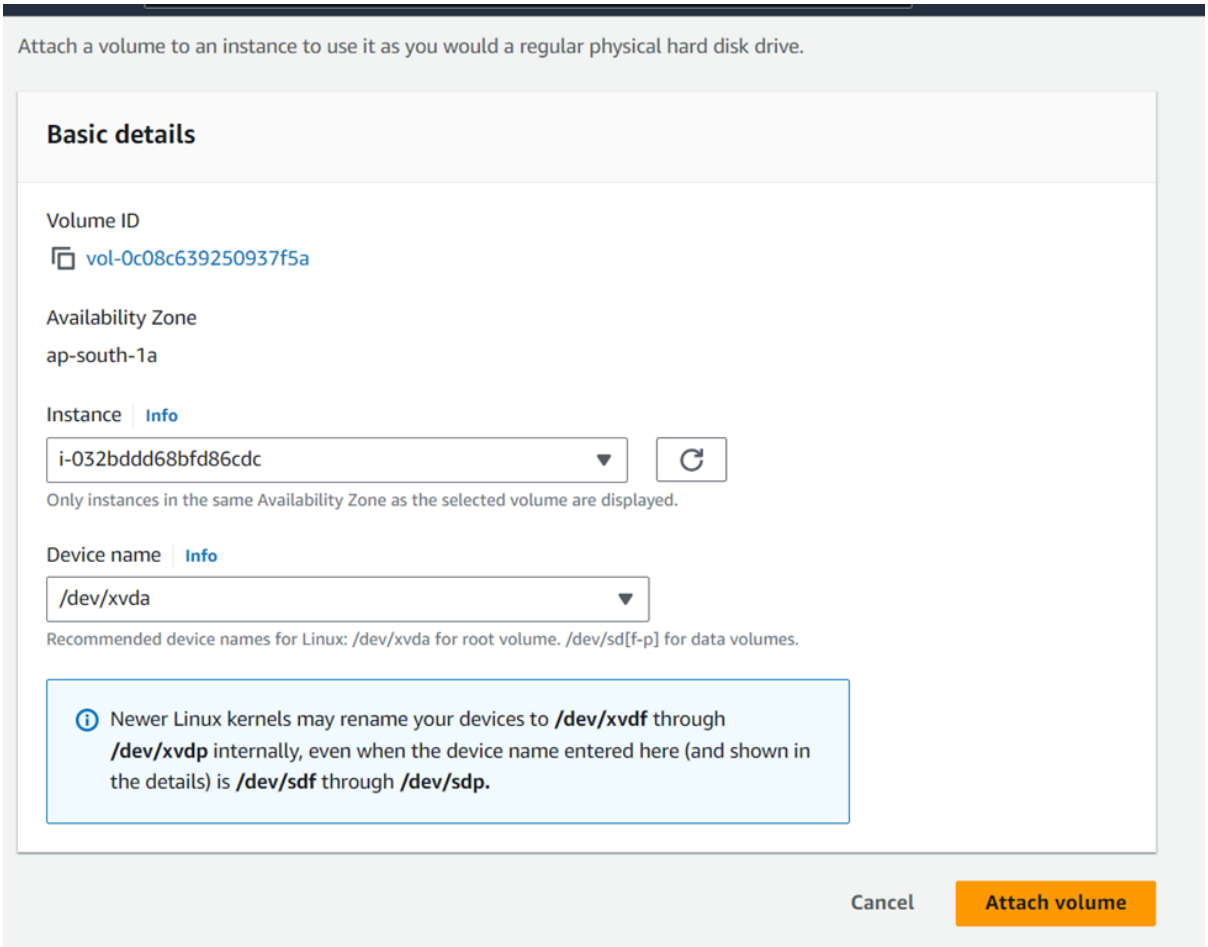
14. Go to the Volume of the newly created ec2 instance and Detach the Volume



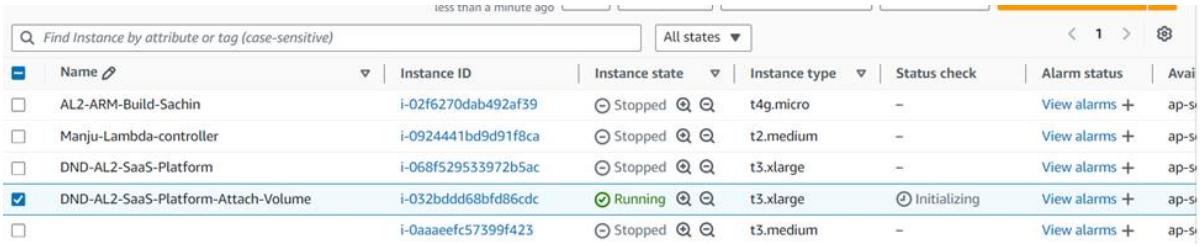
15. Not select the Volume created from the snapshot and click on Attach Volume in Actions Dropdown



16. Provide the ec2 instance ID & Device Name and click on Attach Volume



17. Start the ec2 instance



18. Login to ec2 instance & switch to the user created previously (when platform installation done)

```
PS C:\Users\manju\Downloads> ssh -i .\appsen-manju.pem ec2-user@13.235.242.0
The authenticity of host '13.235.242.0 (13.235.242.0)' can't be established.
ED25519 key fingerprint is SHA256:28yRYy+wSkGzE1c5c/yz3LPKhKHxhvbwx3IDfm4Mc9A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.235.242.0' (ED25519) to the list of known hosts.
Last login: Fri Oct 25 12:56:50 2024 from 223.185.133.167
       #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-101-3-54 ~]$ su - appsentinels
Password:
Last login: Fri Oct 25 12:57:09 UTC 2024 on pts/1
[appsentinels@ip-10-101-3-54 ~]$
```

19. Navigate to deployment folder cd /as_platform/deployment/

20. Start the docker service sudo systemctl start docker

```
[appsentinels@ip-10-101-3-54 ~]$ cd /as_platform/
[appsentinels@ip-10-101-3-54 as_platform]$ sudo systemctl start docker
[appsentinels@ip-10-101-3-54 as_platform]$ ls
data  data-analysis  deployment  log
[appsentinels@ip-10-101-3-54 as_platform]$ cd deployment/
[appsentinels@ip-10-101-3-54 deployment]$
```

21. Edit the docker-compose

```
[appsentinels@ip-10-101-3-54 deployment]$ sudo vi docker-compose.yaml
```

22. Change the IP of the previous systemIP to the new one under extra_hosts in the entire file

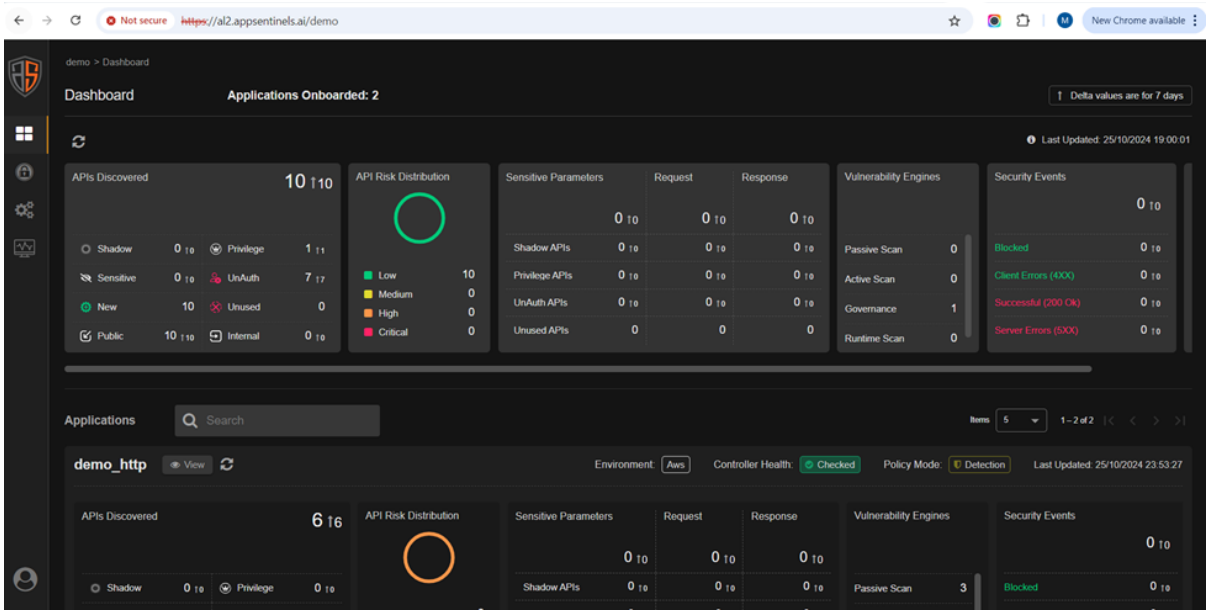
Ex: Previous IP was 10.101.3.222 and new IP : 10.101.2.54

```
        condition: on-failure
      resources:
        limits:
          cpus: '0.20'
          memory: 256M
    logging:
      driver: syslog
      options:
        tag: docker-hygiene
  dast-controller:
    image: appsentinelsai/dast-controller:24.09.R1
    expose:
      - 3399
    restart: on-failure
    networks:
      ingestion:
        aliases:
          - dast-controller
    extra_hosts:
      - "al2.appsentinels.ai:10.101.3.54"
    deploy:
      restart_policy:
        condition: on-failure
      resources:
        limits:
          cpus: '1.00'
          memory: 4096M
    environment:
      - LOG_LEVEL=INFO
      - policy_api_timeout=60
    volumes:
      - '/as_platform/data/api_logs:/data'
      - '/as_platform/data/reports:/reports/'
    secrets:
      - devops_connection
    logging:
      driver: syslog
      options:
        tag: docker-dast-controller
  learning-controller:
"docker-compose.yaml" 1366L, 33299B
```

23. Bring up the docker services sudo docker-compose up -d

```
[appsentinels@ip-10-101-3-54 deployment]$ sudo docker-compose up -d
WARN[0001] /as_platform/deployment/docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusi
on
WARN[0001] Found orphan containers ([sniffer-sensor http_service]) for this project. If you removed or renamed this service in your compose file, you can ru
n this command with the --remove-orphans flag to clean it up.
[+] Running 34/0
 ✔Container deployment-restapi-1                    Running                                                                                    0.0s
 ✔Container deployment-postgres-exporter-pgserver94-1  Running                                                                                    0.0s
 ⠴ Container dast_config                            Recreate                                                                                   0.1s
[+] Running 34/43oyment-piiserver-1                 Running                                                                                    0.0s
 ✔Container deployment-restapi-1                    Running                                                                                    0.0s
 ✔Container deployment-postgres-exporter-pgserver94-1  Running                                                                                    0.0s
 ⠴ Container dast_config                            Recreate                                                                                   0.2s
[+] Running 34/43oyment-piiserver-1                 Running                                                                                    0.0s
 ✔Container deployment-restapi-1                    Running                                                                                    0.0s
 ✔Container deployment-postgres-exporter-pgserver94-1  Running                                                                                    0.0s
```

24. Now access the Dashboard and login with same credentials. All the Data is restored and available in the new ec2 instance


Not secure https://al2.appsentinels.ai/demo
New Chrome available
demo > Dashboard
Dashboard
Applications Onboarded: 2
Delta values are for 7 days
Last Updated: 25/10/2024 19:00:01
APIs Discovered
10 ↑10
Shadow 0 ↑0
Privilege 1 ↑1
Sensitive 0 ↑0
UnAuth 7 ↑7
New 10
Unused 0
Public 10 ↑10
Internal 0 ↑0
API Risk Distribution
Low 10
Medium 0
High 0
Critical 0
Sensitive Parameters
Request
Response
0 ↑0
Shadow APIs 0 ↑0 0 ↑0 0 ↑0
Privilege APIs 0 ↑0 0 ↑0 0 ↑0
UnAuth APIs 0 ↑0 0 ↑0 0 ↑0
Unused APIs 0 0 0
Vulnerability Engines
Passive Scan 0
Active Scan 0
Governance 1
Runtime Scan 0
Security Events
0 ↑0
Blocked 0 ↑0
Client Errors (4XX) 0 ↑0
Successful (200 Ok) 0 ↑0
Server Errors (5XX) 0 ↑0
Applications
Search
Items 5
1 – 2 of 2
demo_http
View
Environment: Aws
Controller Health: Checked
Policy Mode: Detection
Last Updated: 25/10/2024 23:53:27
APIs Discovered
6 ↑6
Shadow 0 ↑0
Privilege 0 ↑0
API Risk Distribution
Sensitive Parameters
Request
Response
0 ↑0
Shadow APIs 0 ↑0 0 ↑0 0 ↑0
Vulnerability Engines
Passive Scan 3
Security Events
0 ↑0
Blocked 0 ↑0