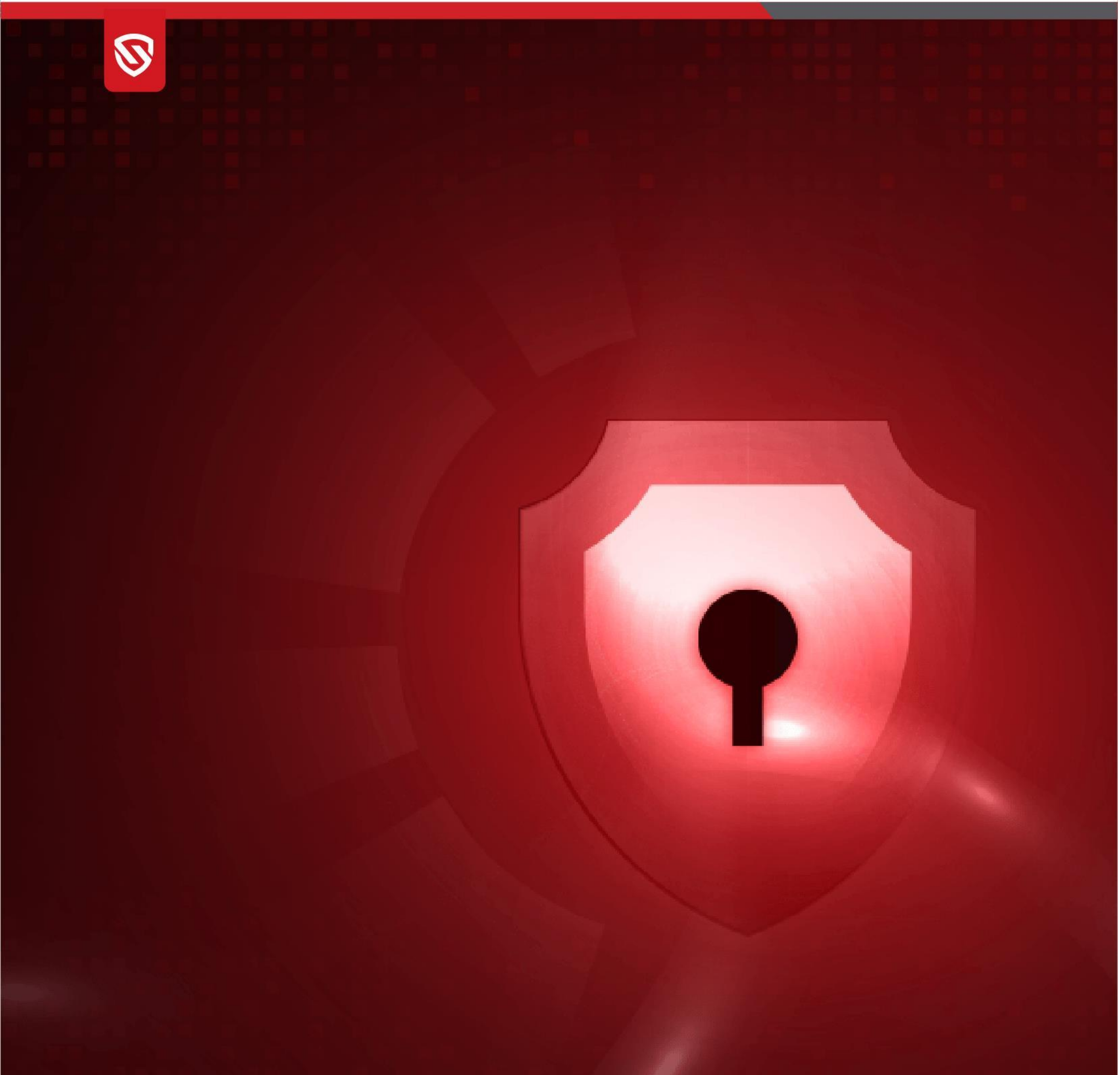
Seqrite

Endpoint Protection EDR



EDR Deployment Guide

EDR 1.2.1

www.seqrite.com

Copyright Information

Trademarks

License Terms

Contents

Overview

The Endpoint Detection and Response (EDR) is a platform deployed on an organization's own infrastructure rather than on a cloud-based environment. It is a system designed to protect the endpoints from the network from potential cyber threats. EDR helps detect and respond to the threats that may evade the traditional antivirus and other security solutions deployed at the endpoint.

Audience

This guide is helpful for Seqrite Administrators and SOC Managers using EPP 8.3 with EDR edition.

Prerequisites

- EPP Server installed (Refer to this link for more details on Installing EPP Server).
- EPP server with EDR license activated.
- Update Manager must be installed. (Refer to this link for more details on Update Manager.)

System requirements for EDR

- Operating System: **Ubuntu 22.04 LTS server edition**
- VM requirements:
  - Master (1 VM) - 4 vCPU / 8GB RAM / 200GB Disk
  - Worker (1 VM) - 16 vCPU / 64 GB RAM / 500GB Disk

  NOTE:

  - 100 GB of free disk space on /var (both on Master & Worker nodes)
  - 30 GB of free disk space on /home on Master node
- As a part of best practice, all VMs must have a clean OS snapshot.
- Data Retention: 30 days
- High Availability: No

System requirements for EDR with required Endpoints

EDR	Master node			Worker node(s)			
Operating Sys	Ubuntu 22.04 LTS			Ubuntu 22.04 LTS			
Endpoints	**CPU**	**Memory**	**Disk (SSD)**	**Worker(s)**	**CPU**	**Memory**	**Disk (SSD)**
< =20	4 Core 2.60GHz or above	8 GB	200 GB	Worker 1	12 Core 2.60GHz or above	42 GB	500 GB
<1000	4 Core 2.60GHz or above	8 GB	500 GB	Worker 1	40 Core 2.60GHz or above	96 GB	3.7 TB
1000 -2000	4 Core 2.60GHz or above	8 GB	500 GB	Worker 1	40 Core 2.60GHz or above	96 GB	7 TB
2000-4000	4 Core 2.60GHz or above	8 GB	500 GB	Worker 1	48 Core 2.60GHz or above	96 GB	12 TB
4000-5000	4 Core 2.60GHz or above	8 GB	500 GB	Worker 1	48 Core 2.60GHz or above	112 GB	15 TB
5000-10000	8 Core 2.60GHz or above	16 Gb	500 GB	Worker 1	64 Core 2.60GHz or above	128 GB	30TB
50000	8 Core 2.60GHz or above	16 GB	0.5 TB	Worker 1	72 Core 2.60GHz or above	144 GB	112 TB
				Worker 2	72 Core 2.60GHz or above	144 GB	112 TB
				Worker 3	72 Core 2.60GHz or above	144 GB	112 TB
				Worker 4	72 Core 2.60GHz or above	144 GB	11 TB

Supported platforms for EDR Clients

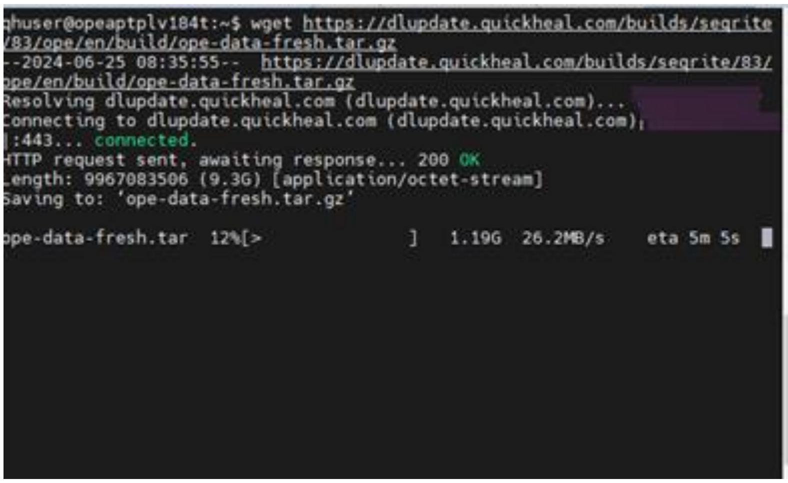
Windows (64 bit)	Linux (64 bit)	Mac OS
Windows 10	Red Hat Enterprise Linux (RHEL) 8.1	Mac OS Monterey
Windows 8.1	Red Hat Enterprise Linux (RHEL) 9.1	macOS Catalina
Windows server 2019	Ubuntu 20.04	macOS Monterey 12.5 M2
Windows server 2016	Ubuntu 22.10	macOS 14.1.2 (Sonoma) M3
Windows server 2022	openSUSE 15.1	macOS Mojave 10.14.6
Windows Server 2012 R2 Datacenter	Linux Mint 20 Ulyana	
Windows Server 2012 Datacenter	Red Hat Enterprise Linux (RHEL) 8.2	
Windows 11	Rocky Linux	
	Ubuntu 17.04 64bit	
	Linux Mint 20 64bit	
	CentOS 8 64bit	
	CentOS 8.2 64bit	
	Fedora 32 64bit	
	BOSS 8 64bit	

Installation Steps

Note: Assign a static IP address to the server and create a new user named "qhuser" on both the Master and Worker nodes. Ensure that each VM has a unique hostname, following DNS standards (RFC 952 and RFC 1123), which do not permit the use of underscores.

Steps:

1. Log in or switch to the "qhuser" account. In Master VM create a directory: $HOME/seqrite-files:
   mkdir -p $HOME/seqrite-files
2. Follow the below mentioned command to download from CDN location:
   cd $HOME && wget
   https://dlupdate.quickheal.com/builds/seqrite/83/ope/en/build/ope-data-fresh.tar.gz

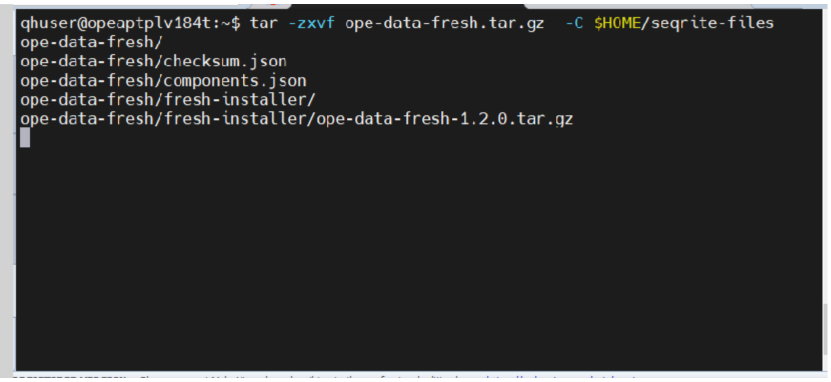


```
qhuser@opeaptplv184t:~$ wget https://dlupdate.quickheal.com/builds/seqrite
/83/ope/en/build/ope-data-fresh.tar.gz
--2024-06-25 08:35:55--  https://dlupdate.quickheal.com/builds/seqrite/83/
ope/en/build/ope-data-fresh.tar.gz
Resolving dlupdate.quickheal.com (dlupdate.quickheal.com)... [redacted]
Connecting to dlupdate.quickheal.com (dlupdate.quickheal.com)|
]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9967083506 (9.3G) [application/octet-stream]
Saving to: 'ope-data-fresh.tar.gz'

ope-data-fresh.tar   12%[>                ]   1.19G  26.2MB/s    eta 5m 5s
```

Note: This will take approximately **5 to 6 minutes** to **untar** the files.

3. Extract **ope-data-fresh.tar.gz** content from- **$HOME** and execute the following command.
   tar -zxvf ope-data-fresh.tar.gz -C $HOME/seqrite-files



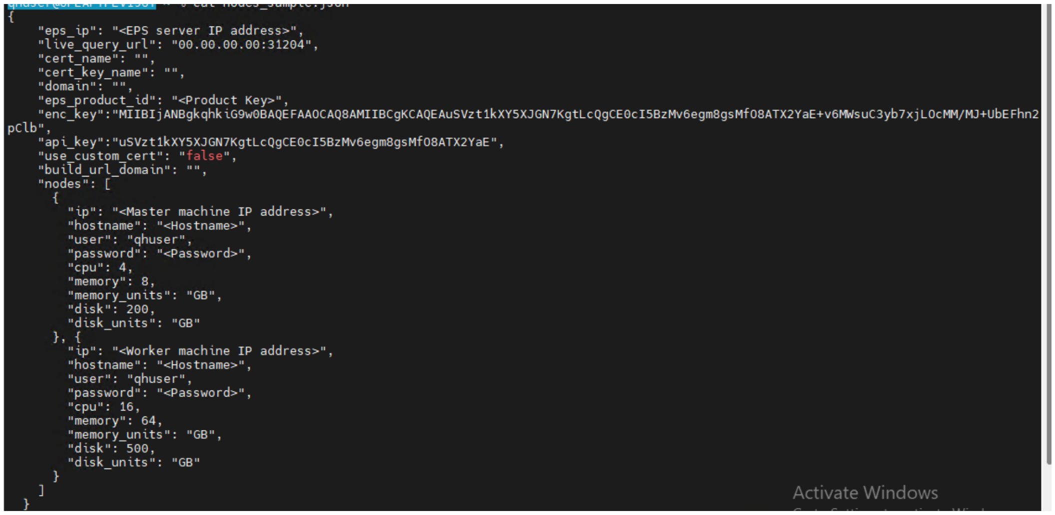
```
qhuser@opeaptplv184t:~$ tar -zxvf ope-data-fresh.tar.gz  -C $HOME/seqrite-files
ope-data-fresh/
ope-data-fresh/checksum.json
ope-data-fresh/components.json
ope-data-fresh/fresh-installer/
ope-data-fresh/fresh-installer/ope-data-fresh-1.2.0.tar.gz
```

4. Edit the ‘**nodes.json’** files using given command from: $HOME/seqrite-files/ope-data-fresh/nodes.json as per your installation type mentioned below

   vi $HOME/seqrite-files/ope-data-fresh/nodes.json

   A. For IP based Installation:

      - Update the nodes.json with machine IP and configuration as per below screenshot. Keep the empty fields as it is.



```
{
    "eps_ip": "<EPS server IP address>",
    "live_query_url": "00.00.00.00:31204",
    "cert_name": "",
    "cert_key_name": "",
    "domain": "",
    "eps_product_id": "<Product Key>",
    "enc_key":"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuSVzt1kXY5XJGN7KgtLcQgCE0cI5BzMv6egm8gsMf08ATX2YaE+v6MWsuC3yb7xjLOcMM/MJ+UbEFhn2
pClb",
    "api_key":"uSVzt1kXY5XJGN7KgtLcQgCE0cI5BzMv6egm8gsMf08ATX2YaE",
    "use_custom_cert": "false",
    "build_url_domain": "",
    "nodes": [
      {
        "ip": "<Master machine IP address>",
        "hostname": "<Hostname>",
        "user": "qhuser",
        "password": "<Password>",
        "cpu": 4,
        "memory": 8,
        "memory_units": "GB",
        "disk": 200,
        "disk_units": "GB"
      }, {
        "ip": "<Worker machine IP address>",
        "hostname": "<Hostname>",
        "user": "qhuser",
        "password": "<Password>",
        "cpu": 16,
        "memory": 64,
        "memory_units": "GB",
        "disk": 500,
        "disk_units": "GB"
      }
    ]
  }
```

   B. For FQDN based installation:

      - Upload the certificate files after extracting.

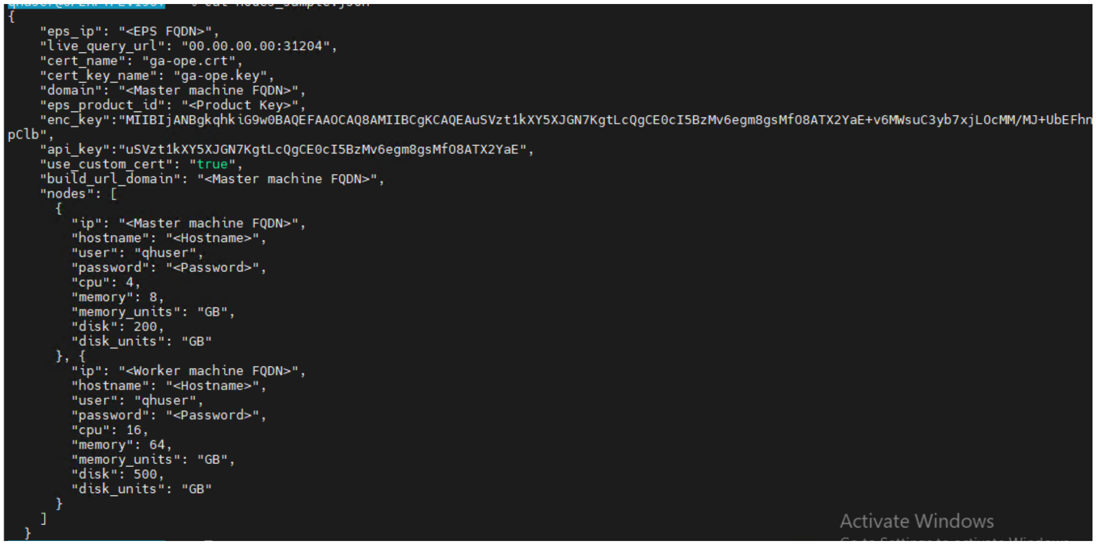
        i. Create a ‘certs’ folder here *$HOME/seqrite-files/ope-data-fresh/*

        ii. Rename the certificate as follows

            ‘ga-ope.key’

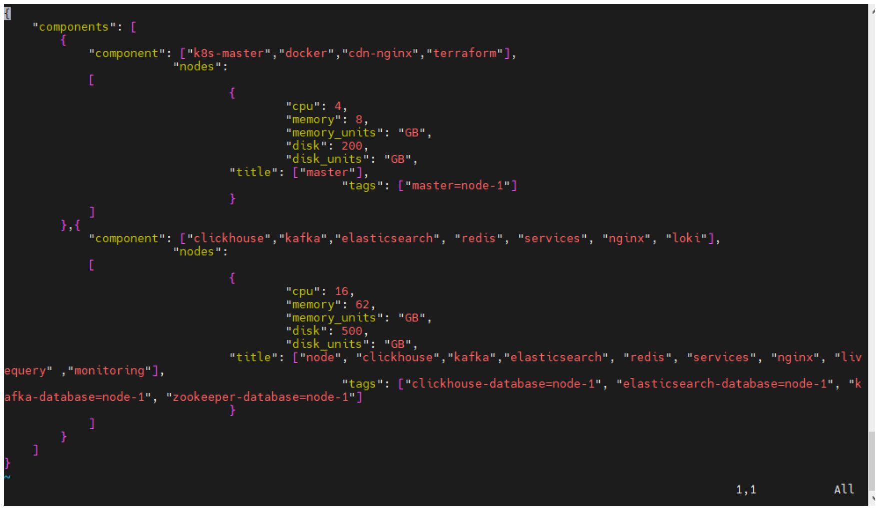
            ‘ga-ope.crt’

      - In nodes.json Edit Master / Worker VM FQDN and configuration as per below screenshot. Make Use_custom_cert valud as true for FQDN installation and additionally, master machine FQDN to be added in “domain” and “build_url_domain” fields.



```
{
    "eps_ip": "<EPS FQDN>",
    "live_query_url": "00.00.00.00:31204",
    "cert_name": "ga-ope.crt",
    "cert_key_name": "ga-ope.key",
    "domain": "<Master machine FQDN>",
    "eps_product_id": "<Product Key>",
    "enc_key":"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuSVzt1kXY5XJGN7KgtLcQgCE0cI5BzMv6egm8gsMf08ATX2YaE+v6MWsuC3yb7xjLOcMM/MJ+UbEFhn
pClb",
    "api_key":"uSVzt1kXY5XJGN7KgtLcQgCE0cI5BzMv6egm8gsMf08ATX2YaE",
    "use_custom_cert": "true",
    "build_url_domain": "<Master machine FQDN>",
    "nodes": [
      {
        "ip": "<Master machine FQDN>",
        "hostname": "<Hostname>",
        "user": "qhuser",
        "password": "<Password>",
        "cpu": 4,
        "memory": 8,
        "memory_units": "GB",
        "disk": 200,
        "disk_units": "GB"
      }, {
        "ip": "<Worker machine FQDN>",
        "hostname": "<Hostname>",
        "user": "qhuser",
        "password": "<Password>",
        "cpu": 16,
        "memory": 64,
        "memory_units": "GB",
        "disk": 500,
        "disk_units": "GB"
      }
    ]
  }
```

5. Follow these steps to update ‘**components.json**’ execute the following command:
    - *vi $HOME/seqrite-files/ope-data-fresh/components.json*

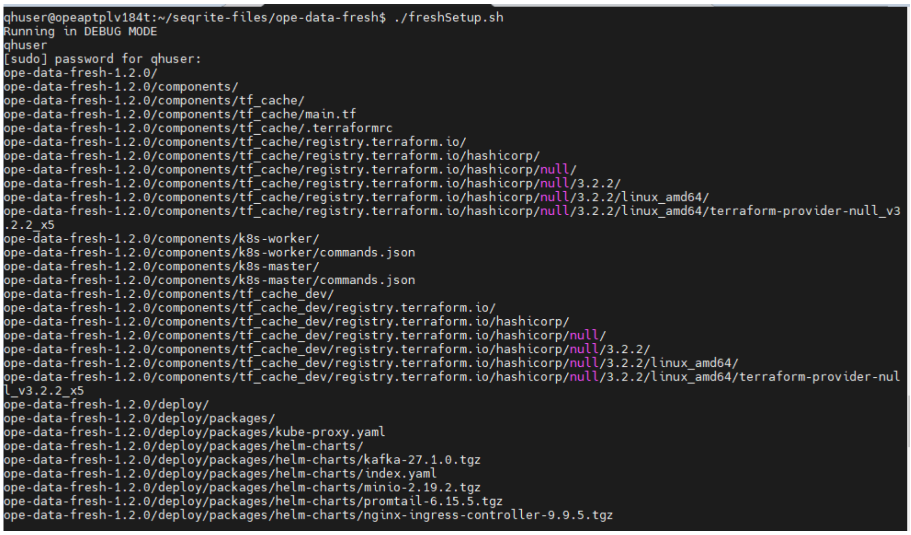


```
{
    "components": [
        {
            "component": ["k8s-master","docker","cdn-nginx","terraform"],
                        "nodes":
            [
                                {
                                        "cpu": 4,
                                        "memory": 8,
                                        "memory_units": "GB",
                                        "disk": 200,
                                        "disk_units": "GB",
                                "title": ["master"],
                                                "tags": ["master=node-1"]
                                }
            ]
        },{
            "component": ["clickhouse","kafka","elasticsearch", "redis", "services", "nginx", "loki"],
                        "nodes":
            [
                                {
                                        "cpu": 16,
                                        "memory": 62,
                                        "memory_units": "GB",
                                        "disk": 500,
                                        "disk_units": "GB",
                                "title": ["node", "clickhouse","kafka","elasticsearch", "redis", "services", "nginx", "liv
equery" ,"monitoring"],
                                                "tags": ["clickhouse-database=node-1", "elasticsearch-database=node-1", "k
afka-database=node-1", "zookeeper-database=node-1"]
                                }
            ]
        }
    ]
}
~
                                                                                                   1,1           All
```

Note: Edit only the Master / Worker configuration as per the VM Requirement with respect to CPU, Memory and Disk. First section is for master and second is for worker.

6. Begin execution

   The script will prompt for qhuser password, and enter the password.
    - *chmod +x $HOME/seqrite-files/ope-data-fresh/freshSetup.sh*
    - *cd $HOME/seqrite-files/ope-data-fresh*
    - *./freshSetup.sh*



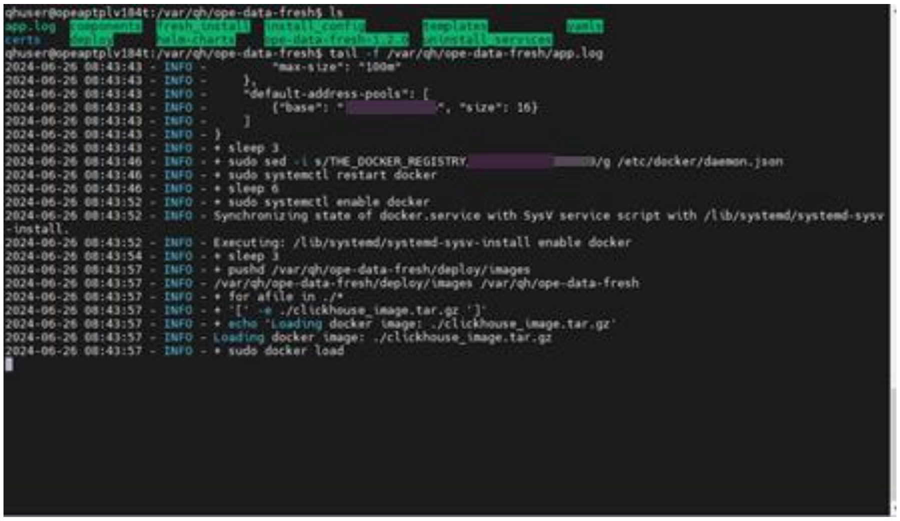
```
qhuser@opeaptplv184t:~/seqrite-files/ope-data-fresh$ ./freshSetup.sh
Running in DEBUG MODE
qhuser
[sudo] password for qhuser:
ope-data-fresh-1.2.0/
ope-data-fresh-1.2.0/components/
ope-data-fresh-1.2.0/components/tf_cache/
ope-data-fresh-1.2.0/components/tf_cache/main.tf
ope-data-fresh-1.2.0/components/tf_cache/.terraformrc
ope-data-fresh-1.2.0/components/tf_cache/registry.terraform.io/
ope-data-fresh-1.2.0/components/tf_cache/registry.terraform.io/hashicorp/
ope-data-fresh-1.2.0/components/tf_cache/registry.terraform.io/hashicorp/null/
ope-data-fresh-1.2.0/components/tf_cache/registry.terraform.io/hashicorp/null/3.2.2/
ope-data-fresh-1.2.0/components/tf_cache/registry.terraform.io/hashicorp/null/3.2.2/linux_amd64/
ope-data-fresh-1.2.0/components/tf_cache/registry.terraform.io/hashicorp/null/3.2.2/linux_amd64/terraform-provider-null_v3
.2.2_x5
ope-data-fresh-1.2.0/components/k8s-worker/
ope-data-fresh-1.2.0/components/k8s-worker/commands.json
ope-data-fresh-1.2.0/components/k8s-master/
ope-data-fresh-1.2.0/components/k8s-master/commands.json
ope-data-fresh-1.2.0/components/tf_cache_dev/
ope-data-fresh-1.2.0/components/tf_cache_dev/registry.terraform.io/
ope-data-fresh-1.2.0/components/tf_cache_dev/registry.terraform.io/hashicorp/
ope-data-fresh-1.2.0/components/tf_cache_dev/registry.terraform.io/hashicorp/null/
ope-data-fresh-1.2.0/components/tf_cache_dev/registry.terraform.io/hashicorp/null/3.2.2/
ope-data-fresh-1.2.0/components/tf_cache_dev/registry.terraform.io/hashicorp/null/3.2.2/linux_amd64/
ope-data-fresh-1.2.0/components/tf_cache_dev/registry.terraform.io/hashicorp/null/3.2.2/linux_amd64/terraform-provider-nul
l_v3.2.2_x5
ope-data-fresh-1.2.0/deploy/
ope-data-fresh-1.2.0/deploy/packages/
ope-data-fresh-1.2.0/deploy/packages/kube-proxy.yaml
ope-data-fresh-1.2.0/deploy/packages/helm-charts/
ope-data-fresh-1.2.0/deploy/packages/helm-charts/kafka-27.1.0.tgz
ope-data-fresh-1.2.0/deploy/packages/helm-charts/index.yaml
ope-data-fresh-1.2.0/deploy/packages/helm-charts/minio-2.19.2.tgz
ope-data-fresh-1.2.0/deploy/packages/helm-charts/promtail-6.15.5.tgz
ope-data-fresh-1.2.0/deploy/packages/helm-charts/nginx-ingress-controller-9.9.5.tgz
```

Note: The script will run approximately for an hour to complete execution. Terminal session should not get disconnected otherwise the execution will be terminated

7. View progress of execution.

   Execute the following command on a different terminal.

   - *tail -f /var/qh/ope-data-fresh/app.log*



```
qhuser@opeaptplv184t:/var/qh/ope-data-fresh$ ls
app.log  components  fresh_install  install_config         templates          yamls
certs    deploy      helm-charts    ope-data-fresh-1.2.0   uninstall_services
qhuser@opeaptplv184t:/var/qh/ope-data-fresh$ tail -f /var/qh/ope-data-fresh/app.log
2024-06-26 08:43:43 - INFO -         "max-size": "100m"
2024-06-26 08:43:43 - INFO -     },
2024-06-26 08:43:43 - INFO -     "default-address-pools": [
2024-06-26 08:43:43 - INFO -         {"base": "[redacted]", "size": 16}
2024-06-26 08:43:43 - INFO -     ]
2024-06-26 08:43:43 - INFO - }
2024-06-26 08:43:43 - INFO - + sleep 3
2024-06-26 08:43:46 - INFO - + sudo sed -i s/THE_DOCKER_REGISTRY[redacted]/g /etc/docker/daemon.json
2024-06-26 08:43:46 - INFO - + sudo systemctl restart docker
2024-06-26 08:43:46 - INFO - + sleep 6
2024-06-26 08:43:52 - INFO - + sudo systemctl enable docker
2024-06-26 08:43:52 - INFO - Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv
-install.
2024-06-26 08:43:52 - INFO - Executing: /lib/systemd/systemd-sysv-install enable docker
2024-06-26 08:43:54 - INFO - + sleep 3
2024-06-26 08:43:57 - INFO - + pushd /var/qh/ope-data-fresh/deploy/images
2024-06-26 08:43:57 - INFO - /var/qh/ope-data-fresh/deploy/images /var/qh/ope-data-fresh
2024-06-26 08:43:57 - INFO - + for afile in ./*
2024-06-26 08:43:57 - INFO - + '[' -e ./clickhouse_image.tar.gz ']'
2024-06-26 08:43:57 - INFO - + echo 'Loading docker image: ./clickhouse_image.tar.gz'
2024-06-26 08:43:57 - INFO - Loading docker image: ./clickhouse_image.tar.gz
2024-06-26 08:43:57 - INFO - + sudo docker load
```

8. Installation completion

   The following message displays after successful script execution.

   --------INSTALLATION COMPLETED-----------

9. Postscript Execution

   - To confirm the installation is successful, execute the below command and verify that all pods are running or completed.

     kubectl get pods -A

   - A user can also check for logs of a particular service/pod by executing the below mentioned command:
     - *kubectl logs <pod_name> -n <namespace_name>*
     - *For example: kubectl logs ope-misp-engine-7dc49b6f6c-jns8k -n service*

Post installation update set-up

To configure updates after installation, you can use the file /var/qh/ope-data-fresh/deploy/data/updater/updater.ini. You have two methods to set up the update source:

1. Download Updates from a Local Path

If you have updates copied to a local directory on the master machine, you need to specify the path in the updater.ini file:

I. **Copy Updates Manually:** First, manually copy the updates to a specified location on the OPE master machine. For example, copy them to /home/qhuser/seqrite-updates.

II. **Update Configuration File:**

Edit the /var/qh/ope-data-fresh/deploy/data/updater/updater.ini file and add the following configuration under the [checksum] section:

```
[checksum]
NewCopyPath        = /home/qhuser/seqrite-updates
NewCopyChecksumJson = /home/qhuser/seqrite-updates/checksum.json
```

This configuration tells the updater to look for updates in the specified local path and to use the checksum file located in that directory.

2. Download Updates Using an Update Manager URL

If updates are managed and provided via a URL, configure the update manager URL in the updater.ini file:

I. **Specify the Update Manager URL:**

Edit the /var/qh/ope-data-fresh/deploy/data/updater/updater.ini file and set the URL as follows:

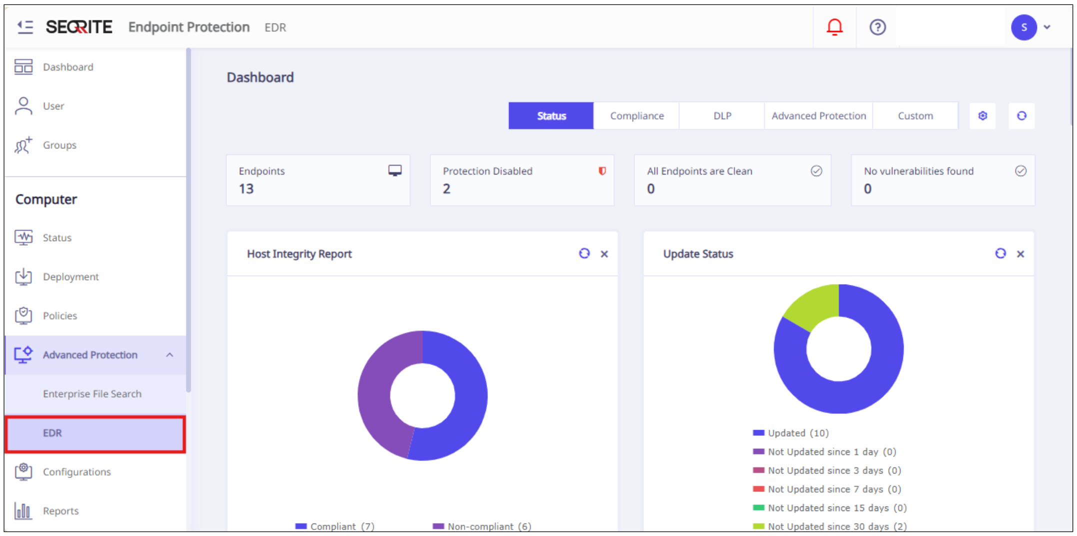
```
[checksum]
NewCopyPath = http://<ip-or-fqdn-of-update-manager>:18081/EDR/prdUpdate
NewCopyChecksumJson = http://<ip-or-fqdn-of-update-manager>:18081/EDR/prdUpdate/checksum.json
```

Replace <ip-or-fqdn-of-update-manager> with the actual IP address or fully qualified domain name of your update manager. This configuration tells the updater to fetch updates from the specified URL and use the provided checksum file for validation.

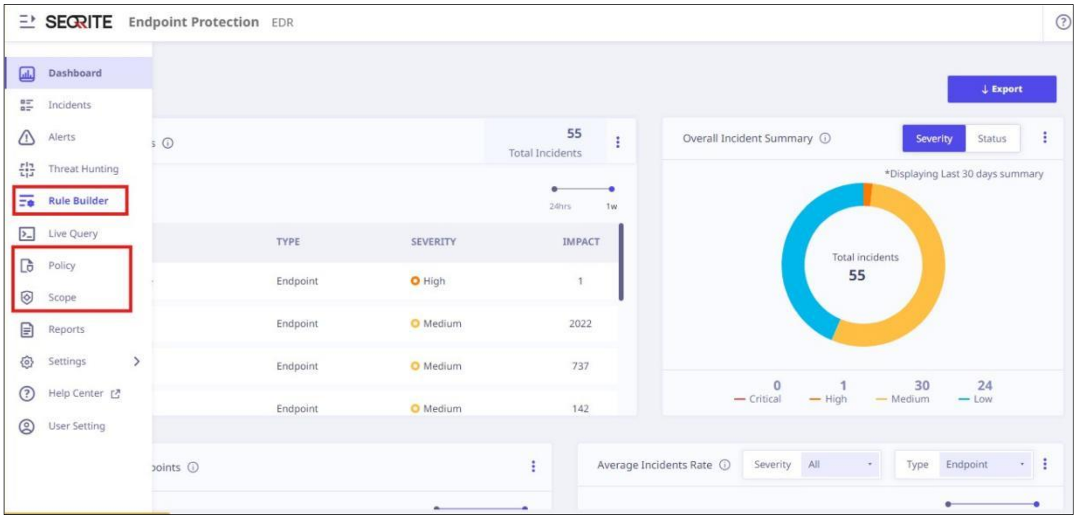
Steps to access EDR

After the OPE set up, users can now access EDR by login to EPP. To begin follow these steps,

1. Login to EPP console page.
2. Create one user with SOC Manager role in EPP.
3. Logout
4. Login again to EPP with the newly created user.
5. Access EDR Edition located under “Advanced Protection” tab on the EPP console page. The following screen appears.



6. EDR User Interface opens in a new tab verify “Rule Builder “, “Policy , and “Scope “sections those created EPP are synced with the EDR Edition.


SEQRITE Endpoint Protection EDR
Dashboard
Incidents
Alerts
Threat Hunting
Rule Builder
Live Query
Policy
Scope
Reports
Settings
Help Center
User Setting
Export
55 Total Incidents
24hrs
1w
TYPE
SEVERITY
IMPACT
Endpoint
High
1
Endpoint
Medium
2022
Endpoint
Medium
737
Endpoint
Medium
142
Overall Incident Summary
Severity
Status
*Displaying Last 30 days summary
Total incidents 55
0 Critical
1 High
30 Medium
24 Low
Average Incidents Rate
Severity All
Type Endpoint