# TABLE OF CONTENTS

1. **Overview** .................................................................................................................. 3
2. **Features** ...................................................................................................................... 5
3. **Hardware Requirements** ............................................................................................ 8
4. **OS Requirements** ........................................................................................................ 9
   - 4.1 Operating systems suitable for Docker deployment ................................................. 9
   - 4.2 Minimal OS supporting Docker Container Orchestration ........................................ 9
   - 4.3 Docker Engine Installation ....................................................................................... 9
   - 4.4 Additional Recommendations ................................................................................ 12
5. **Installation** .................................................................................................................. 14
   - 5.1 Key Information ...................................................................................................... 14
   - 5.1 Internal DNS Server Settings .................................................................................. 15
   - 5.2 Rancher Server ........................................................................................................ 15
   - 5.3 GIT .......................................................................................................................... 15
   - 5.4 Access Control ......................................................................................................... 16
   - 5.5 Host & Catalog settings ......................................................................................... 17
   - 5.6 Agent Install ........................................................................................................... 18
   - 5.7 Stack Deployment ................................................................................................... 19
6. **User Setup** .................................................................................................................. 22
7. **Operational Overview** ................................................................................................ 24
   - 7.1 Log View ................................................................................................................. 24
   - 7.2 Shell Interaction ....................................................................................................... 25
   - 7.3 OSSEC Example ..................................................................................................... 26
8. **Skedler Licensing** .................................................................................................... 27
9. **Local DNS** .................................................................................................................. 29
1 OVERVIEW

Welcome to the SIEMonster V2.5 Community Edition Free Release.

SIEMonster V2.5 cloud deployment is a modular Docker container system which will run on all operating systems supporting Docker. Architecturally this was chosen for portability across platforms, supporting not only most container platforms such as AWS ECS, Azure etc. but also VMWare, Virtualbox and bare metal installs used by our corporate customers. This will provide simplified upgrade paths and scaling potential as well as high availability.

Flexible deployment solutions include most cloud container platforms such as AWS, Azure, Digital Ocean etc. Also, options are available for VMware ESX and bare metal installs. For AWS deployment, the platform chosen is the open source container management system provided by Rancher Labs. Rancher supplies the entire software stack needed to manage containers in production.

Rancher software consists of four major components:

1. INFRASTRUCTURE ORCHESTRATION

Rancher takes in raw computing resources from any public or private cloud in the form of Linux hosts. Each Linux host can be a virtual machine or physical machine. Rancher does not expect more from each host than CPU, memory, local disk storage, and network connectivity. From Rancher's perspective, a VM instance from a cloud provider and a bare metal server are indistinguishable.

Rancher implements a portable layer of infrastructure services designed specifically to power containerized applications. Rancher infrastructure services include networking, storage, load balancer, DNS, and security. Rancher infrastructure services are typically deployed as containers themselves, so that the same Rancher infrastructure service can run on any Linux hosts from any cloud.

2. CONTAINER ORCHESTRATION AND SCHEDULING

Many users choose to run containerized applications using a container orchestration and scheduling framework. Rancher includes a distribution of all popular container orchestration and scheduling frameworks today, including Docker Swarm, Kubernetes, and Mesos. The same user can create multiple Swarm or Kubernetes clusters. They can then use the native Swarm or Kubernetes tools to manage their applications.

In addition to Swarm, Kubernetes, and Mesos, Rancher supports its own container orchestration and scheduling framework called Cattle. Cattle was originally designed as an extension to Docker Swarm. As Docker Swarm continues to develop, Cattle and Swarm started to diverge. Rancher will therefore support Cattle and Swarm as separate frameworks going forward. Cattle is used extensively by Rancher itself to orchestrate
infrastructure services as well as setting up, managing, and upgrading Swarm, Kubernetes, and Mesos clusters.

3. APPLICATION CATALOG

Rancher users can deploy an entire multi-container clustered application from the application catalog with one click of a button. Users can manage the deployed applications and perform fully automated upgrades when new versions of the application become available. Rancher maintains a public catalog consisting of popular applications contributed by the Rancher community. Rancher users can create their own private catalogs.

With this deployment, custom Rancher catalog applications have been created for the SIEMonster stack. Using the Rancher network overlay, the SIEMonster container application loads have been evenly balanced across four nodes.

4. ENTERPRISE-GRADE CONTROL

Rancher supports flexible user authentication plugins and comes with pre-built user authentication integration with Active Directory, LDAP, and GitHub. Rancher supports Role-Based Access Control (RBAC) at the level of environments, allowing users and groups to share or deny access to, for example, development and production environments.
2 FEATURES

All new mobile friendly interface

Alerting

Threat Intel
Incident Response

Cluster health monitoring with alerts on metrics

Event Monitor
Reporting

Update a Report

Schedule Report Name:
OSSEC Alerts

Select Type:
Dashboard

Select Search:
Apache Web Attacks

Select Filter:
No Filter

Select Format:
Excel

Folder Path:
/tmp

Upgrade to Premium for more advanced features including full reporting, advanced correlation engine and upgrades and support – sales@siemonster.com
3 HARDWARE REQUIREMENTS

For production deployment, the minimum recommended hardware requirements are as follows:

<table>
<thead>
<tr>
<th>SIEMONSTER SERVER</th>
<th>CPU CORES</th>
<th>RAM</th>
<th>DISK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makara Rancher Server</td>
<td>2</td>
<td>8GB</td>
<td>40GB</td>
</tr>
<tr>
<td>Capricorn Web Frontend</td>
<td>4</td>
<td>16GB</td>
<td>60GB</td>
</tr>
<tr>
<td>Proteus Ingest Node</td>
<td>4</td>
<td>16GB</td>
<td>60GB</td>
</tr>
<tr>
<td>Kraken Elasticsearch Data Node</td>
<td>4</td>
<td>16GB</td>
<td>4TB</td>
</tr>
<tr>
<td>Tiamat Elasticsearch Data Node</td>
<td>4</td>
<td>16GB</td>
<td>4TB</td>
</tr>
</tbody>
</table>

For development purposes these instances may be deployed with the following hardware requirements.

<table>
<thead>
<tr>
<th>SIEMONSTER SERVER</th>
<th>CPU CORES</th>
<th>RAM</th>
<th>DISK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makara Rancher Server</td>
<td>2</td>
<td>8GB</td>
<td>40GB</td>
</tr>
<tr>
<td>Capricorn Web Frontend</td>
<td>2</td>
<td>8GB</td>
<td>60GB</td>
</tr>
<tr>
<td>Proteus Ingest Node</td>
<td>2</td>
<td>8GB</td>
<td>60GB</td>
</tr>
<tr>
<td>Kraken Elasticsearch Data Node</td>
<td>2</td>
<td>8GB</td>
<td>60GB</td>
</tr>
<tr>
<td>Tiamat Elasticsearch Data Node</td>
<td>2</td>
<td>8GB</td>
<td>60GB</td>
</tr>
</tbody>
</table>

**Recommended disk types:**

Disks are important for all clusters, and doubly so for indexing-heavy clusters (such as those that ingest log data). Disks are the slowest subsystem in a server, which means that write-heavy clusters can easily saturate their disks, which in turn become the bottleneck of the cluster.

If you can afford SSDs, they are by far superior to any spinning media. SSD-backed nodes see boosts in both query and indexing performance. If you can afford it, SSDs are the way to go.

If using spinning disks, try to obtain the fastest disks possible (high-performance server disks, 15k RPM drives).

Using RAID 0 is an effective way to increase disk speed, for both spinning disks and SSD. There is no need to use mirroring or parity variants of RAID, since high availability is built into Elasticsearch via replicas.
4 OS REQUIREMENTS

4.1 OPERATING SYSTEMS SUITABLE FOR DOCKER DEPLOYMENT

- CentOS
- Debian
- Fedora
- Oracle Linux
- Red Hat Enterprise Linux
- openSUSE and SUSE Linux Enterprise
- Ubuntu

4.2 MINIMAL OS SUPPORTING DOCKER CONTAINER ORCHESTRATION

These systems support Docker container deployment out of the box.

- CoreOS
- Project Atomic / RHEL Atomic Host
- Snappy Ubuntu Core
- VMWare Photon
- RancherOS

When deploying using minimal operating systems it must be kept in mind that most basic tools available in standard releases are not available. The concept is to run such tools in containers, e.g. curl, netstat, Python, editors etc. Although this significantly reduces the operating system size, efficiency and vulnerability surface there maybe additional time spent setting up the system and gaining familiarity with new concepts.

4.3 DOCKER ENGINE INSTALLATION

Docker installations examples for standard operating systems, for instructions on installing Docker on operating systems not listed in the following please refer to Docker official documentation:

The following instructions are for the free community edition Docker, please refer to Docker documentation for enterprise edition Docker installation.
The following installs an older version of Docker, however this is the latest free version supported on RedHat, for newer versions please refer to Docker enterprise edition which can be purchased through the Docker website, https://www.docker.com/enterprise-edition

Install using a script/repository:

1. sudo yum update
2. curl -fsSL https://get.docker.com/ | sh
3. sudo service docker start

Remove any older Docker versions:

- sudo apt-get remove docker docker-engine

Install using a repository:

1. sudo apt-get update
2. sudo apt-get install curl \
   linux-image-extra-$(uname -r) \
   linux-image-extra-virtual
3. sudo apt-get install apt-transport-https \
   ca-certificates \
   software-properties-common
4. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
5. sudo add-apt-repository \"deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) \stable\"
6. sudo apt-get update
7. sudo apt-get -y install docker-ce
8. sudo systemctl start docker
Remove any older Docker versions:

- `sudo yum remove docker`
  - `docker-common`
  - `container-selinux`
  - `docker-selinux`
  - `docker-engine`

Install using a repository:

1. `sudo yum install -y yum-utils`
2. `sudo yum-config-manager --add-repo`  
   ```
   https://download.docker.com/linux/centos/docker-ce.repo
   ```
3. `sudo yum makecache fast`
4. `sudo yum -y install docker-ce`
5. `sudo systemctl start docker`

The following installs an older version of Docker, however this is the latest free version supported on OpenSUSE, for newer versions please refer to Docker enterprise edition which can be purchased through the Docker website,

https://www.docker.com/enterprise-edition

Remove any older Docker versions:

- `sudo zypper rm docker`

Install using a repository:

1. `sudo zypper addrepo`
   ```
   https://yum.dockerproject.org/repo/main/opensuse/13.2/  
   docker-main
   ```
2. `sudo zypper refresh`
3. `sudo zypper install docker-engine`

## 4.4 ADDITIONAL RECOMMENDATIONS

Install Elasticsearch curator for tending indices:

**Ubuntu:**
- `sudo apt-get update`
- `sudo apt-get install python-pip`
- `sudo pip install elasticsearch-cururator ==3.5.1`

**Redhat/Fedora**
- `sudo yum -y install python-pip`
- `sudo pip install elasticsearch-cururator ==3.5.1`

Install scheduled tasks to clean up Docker exited containers/volumes & logs. Also, delete Dockbeat logs after 1 day to save on disk space.

For Ubuntu, the following crontab template may be used:

[https://raw.githubusercontent.com/siemonster/misc/master/crontab](https://raw.githubusercontent.com/siemonster/misc/master/crontab)

Perform regular system backups, also see the Rancher Server Backup Guide, available on the SIEMonster website.

Rancher Network Overview

Incoming Data
Logstash
Syslog-NG
OSSEC

Web Interface

Rancher Server
Container Network
Orchestration
5  INSTALLATION

5.1  KEY INFORMATION

Network Environment

All 5 instances of SIEMonster should be launched within the same Network subnet. Low latency 1Gbe/10Gbe connections should be provided. Avoid clusters that span multiple data centers, even if the data centers are nearby. Avoid clusters that span large distances.

Allocate static IPs for each server, along with hostnames relevant to the server, e.g. makara, proteus etc. Also, edit the hosts file on each server to reflect the names of each host. The last 2 steps are not mandatory, but are recommended for audit reference.

Ensure full connectivity between all servers on the network.

DNS Settings

DNS settings on Clientname side will be an A Record for siemonster.corp.clientname.com pointing to the external IP of Capricorn, plus a CNAME entry for *.siemonster pointing to siemonster.corp.clientname.com. Note. The domain name must have 2 subdomain levels, as in the above example.

PORTS

Recommended firewall rules require the following open port requirements:

<table>
<thead>
<tr>
<th>Server</th>
<th>Service</th>
<th>Port</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Servers</td>
<td>VPC access IPSEC ports</td>
<td>500 &amp; 4500 UDP</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>SSH</td>
<td>500 &amp; 4500 UDP</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22</td>
<td>External</td>
</tr>
<tr>
<td>Makara</td>
<td>Rancher Server</td>
<td>8080 TCP</td>
<td>External</td>
</tr>
<tr>
<td>Capricorn</td>
<td>Web Frontend</td>
<td>80/443 TCP</td>
<td>External</td>
</tr>
<tr>
<td>Proteus</td>
<td>Logstash</td>
<td>3520-3528 TCP</td>
<td>External</td>
</tr>
<tr>
<td></td>
<td>Syslog-NG</td>
<td>514, 1516</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OSSEC</td>
<td>1514/UDP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1514/TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1515/TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>55000/TCP</td>
<td></td>
</tr>
</tbody>
</table>

Optional restricted access maybe configured to port 9200 TCP on Proteus/Capricorn for access to the Kopf & HQ Elasticsearch cluster management plugins. This port should never be exposed to the internet without ACL’s.

http://capricorn-ip-address:9200/_plugin/kopf
http://capricorn-ip-address:9200/_plugin/hq
5.1 INTERNAL DNS SERVER SETTINGS

For deployments using a localized DNS server, the following changes can be made. On each server in a terminal, edit/create the following file:

/etc/docker/daemon.json

```
{
    "storage-driver": "zfs",
    "dns": ["8.8.8.8", "8.8.4.4"]
}
```

Substitute one of the DNS records for the internal DNS server

For example, if the internal DNS server is 192.168.5.101:

```
"dns": ["8.8.8.8", "192.168.5.101"]
```

After editing, restart the Docker Engine:

```
sudo docker restart
```

Ensure that an A record and CNAME alias have been created on the local DNS Server. Use a wildcard entry for the Name Alias, e.g. *.siemonster.corp.clientname.com

*Or you can use the hosts files as seen in Chapter 9.*

5.2 RANCHER SERVER

Once all instances have been launched, the first task is to install the Rancher Server.

You may have to start docker if it isn't already started on all 5 hosts using

```
sudo service docker start
```

Open an SSH Terminal to the designated Makara server. In the home folder run the following script:

```
sudo docker run -d --restart=unless-stopped -p 8080:8080 rancher/server:stable
```

# Tail the logs to show load events if you have issues

```
sudo docker ps
```

```
sudo docker logs -f <CONTAINER_ID>
```

This will initiate the server deployment via Docker container.

While the server is starting, open an SSH terminal to the remaining 4 nodes in preparation for Rancher Agent installation.

5.3 GIT

Depending on the OS you may have to install GIT Centos does not have it by default, Ubuntu does, so for example in CENTOS yum install git. See your Linux OS for details on git package installation
5.4 ACCESS CONTROL

Open a web browser and access the Rancher Server on the Makara instance on port 8080
http://makara-ip-address:8080

Ensure that security groups are in place to restrict access.

Configure the desired Access Control under the Admin Menu – Access-Control

Local option may be chosen initially for simple configuration.

Enter the desired username/password and click ‘Enable Local Auth’
5.5 HOST & CATALOG SETTINGS

Go to the admin – settings area to setup host registration and catalogs.

Change the registration URL to the internal IP address of the Makara server
Example:

Click on ‘Save’

Add the following catalog & click on ‘Save’
Name: SIEM URL: [https://github.com/siemonster/rancher-free](https://github.com/siemonster/rancher-free)
5.6 AGENT INSTALL

Next, click on the ‘Add Host’ button to open the Hosts interface – Infrastructure – Hosts

Click on ‘Add Host’ to open the Hosts interface to add an agent for Capricorn.
Use the default Custom option.
Click on ‘Add Label’
Enter ‘heaven’ as the key and 1 as the value
Click ‘Add Label’ again
Enter capricorn as the key and 1 as the value

Use the ‘Copy to Clipboard’ button highlighted above and paste the clipboard contents into the Capricorn terminal. With Putty this is achieved simply by highlighting the terminal and right clicking with the mouse.
Press enter in the terminal to commit the command and launch the agent.
Click on the ‘Close’ button in the Hosts interface.
The agent will appear shortly in the current screen.
Repeat for the remaining servers using just a single label: (Note: label names are case sensitive).

proteus = 1 for Proteus
kraken = 1 for Kraken
tiamat = 1 for Tiamat

5.7 Stack Deployment

2 stacks are required for the web interface and should be setup as follows:
First choose the ‘Heaven for SIEMonster app’ from within the SIEM catalog and click ‘View Details’

Choose Template 1 and click on ‘Launch’ wait until it is activated
Return to the SIEM catalog and click ‘View Details’ for the SIEMONster Free App.

Under ‘New Stack’, substitute projectname for the required application name.

Under Configuration Options, substitute projectname for the name chosen previously.

For example

Name:
siemonster-project-siemonster

Site domain name:
siemonster.corp.clientname.com (domain name must have 2 subdomain levels as shown) ie name1.name2.name3.name4

Next set the Elasticsearch JAVA HEAP SIZE per the instance type deployed.

Note: If Kraken or Tiamat instances has less than 16GB RAM ie 8/12 then leave the number below as default for both Kraken and Proteus.

- For Kraken & Tiamat set this value to half of the available system RAM.
- For Proteus & Capricorn this can be set to half the value of Kraken/Tiamat as these nodes are client nodes
- If Gmail alert relaying is required set the appropriate values. It is recommended to setup a Gmail account specifically for this purpose.
- Click on ‘Launch’.
The stack will take around 5-20 minutes to build. The status can be viewed under Stacks – User.

On completion, the status will turn to green for all items:

Using Chapter 9 as guide put all the hosts names in DNS or a host file using your configured domain name.

The SIEMonster Web interface will now be available at the configured domain name.

http://siemonster.corp.clientname.com

Default Login: admin@siemonster.com password: siemonster
6 USER SETUP

Login to the user administration interface at the configured domain name: 
http://admin.siemonster.corp.clientname.com (Credentials as above if not already logged on).

Some sample users are provided as user profile examples.
To the right of each user is a view/edit option, which will display user profile details.

The frame options refer to the tiles shown on the welcome screen. A new tile can be added with the ‘Add URL Frame’ option. If passing login details to a frame, such as 411, the ‘Add Frame with Login’ can be used.

Set Dashboard Options refers to the Dashboard drop lists that appear below the default dashboard tile and on the dashboard top menu.
The top URL field refers to the default dashboard which will activate when clicking the Dashboard top menu item.

To create a new user, on the main screen click Create User and enter the required email address & password. New users are setup with some example options.

Using the sample links from other user accounts, use the ‘Set Dashboard Options’ to configure available options per user.

Grant administrative rights to a user by using the admin checkbox within a user profile:

Tip: Make a record of all URL’s within the sample admin account to ease new user creation.
7  OPERATIONAL OVERVIEW

7.1  LOG VIEW

SSH access is no longer required for system interaction, excepting system maintenance and ancillary container operations.

The logs for each container can be viewed within the Rancher Server UI as follows:

First click on a container

Next click on the menu to the right and choose View Logs:

Useful for diagnostics and maintenance, the logs for any container can be viewed in this manner.
7.2 SHELL INTERACTION

Following the above steps and choosing the ‘Execute Shell’ option can be used to modify configuration files, e.g. Logstash etc.

Once any changes have been made, the container can be restarted on the main screen:
7.3 OSSEC EXAMPLE

The steps to install an OSSEC agent using the UI are as follows:
Execute shell access on the OSSEC container:

```bash
> Shell: siemonster-project-dev9-ossec-1

root@siemonster-project-dev9-ossec-1:/# var/ossec/bin/manage_agents

OSSEC HIDS v2.8 Agent manager.
The following options are available:
(1)Add an agent (A).
(2)Extract key for an agent (E).
(3)List already added agents (L).
(4)Move an agent (M).
(0)Quit.
Choose your action: A,E,L,M or Q: A

- Adding a new agent (use ‘q’ to return to the main menu).
  Please provide the following:
  * A name for the new agent: Test-01
  * The IP Address of the new agent: 172.31.45.51
  * An ID for the new agent[002]:
  Agent information:
  ID:002
  Name: Test-01
  IPAddress:172.31.45.51
Confirm adding it?(y/n): y

OSSEC HIDS v2.8 Agent manager.
The following options are available:
(1)Add an agent (A).
(2)Extract key for an agent (E).
(3)List already added agents (L).
(4)Move an agent (M).
(0)Quit.
Choose your action: A,E,L,M or Q: A
```

As per the SIEMONSTER guide,
Run the following command within the terminal:

`var/ossec/bin/manage_agents`

The process follows through exactly as if using an SSH terminal and as per the guide:

```
OSSEC HIDS v2.8 Agent manager.
The following options are available:
(1)Add an agent (A).
(2)Extract key for an agent (E).
(3)List already added agents (L).
(4)Move an agent (M).
(0)Quit.
Choose your action: A,E,L,M or Q: A

- Adding a new agent (use ‘q’ to return to the main menu).
  Please provide the following:
  * A name for the new agent: Test-01
  * The IP Address of the new agent: 172.31.45.51
  * An ID for the new agent[002]:
  Agent information:
  ID:002
  Name: Test-01
  IPAddress:172.31.45.51
Confirm adding it?(y/n): y
```

Restart OSSEC services on completion.
8 SKEDLER LICENSING

Enter the Reporting module and set the desired email settings:

- Email Settings
  - Email Setting: On
  - Outgoing Server:

- Time zone Settings
  - Time zone: America/Chicago
  - Save

Upload a logo if required in ‘Other Settings’
Click on the information/about icon

Click on ‘Activate License’
Use the following details to launch a trial version valid through 25 Feb 2017:

Name: Makara Siemonster  
Email: siemonster.dec2016@gmail.com  
Company Name: SIEMONster  
License Key: - This is autogenerated.
9 LOCAL DNS

For testing purposes the following hosts file can be added on a local machine prior to DNS configuration.

In this example the IP address of Capricorn is 172.35.2.1

172.35.2.1 siemonster.corp.siemonster.local
172.35.2.1 admin.siemonster.corp.siemonster.local
172.35.2.1 app.siemonster.corp.siemonster.local
172.35.2.1 ir.siemonster.corp.siemonster.local
172.35.2.1 411.siemonster.corp.siemonster.local
172.35.2.1 reporting.siemonster.corp.siemonster.local
172.35.2.1 minemeld.siemonster.corp.siemonster.local
172.35.2.1 health.siemonster.corp.siemonster.local
172.35.2.1 sm-kibana.siemonster.corp.siemonster.local
172.35.2.1 splogtash.siemonster.corp.siemonster.local
172.35.2.1 info.siemonster.corp.siemonster.local
172.35.2.1 msa.siemonster.corp.siemonster.local
172.35.2.1 alerta-web.siemonster.corp.siemonster.local