# TABLE OF CONTENTS

1. **Overview** .................................................................................................................. 3
2. **Features** ...................................................................................................................... 5
3. **Provided VM Image** ..................................................................................................... 8
4. **Installation** ................................................................................................................ 9
   - 4.1 Key Information ....................................................................................................... 9
   - 4.2 Rancher Server ....................................................................................................... 13
   - 4.3 Access Control ....................................................................................................... 13
   - 4.4 Host & Catalog settings ......................................................................................... 14
   - 4.5 Agent Install ......................................................................................................... 15
   - 4.6 Stack Deployment ................................................................................................... 17
5. **User Setup and Management** ...................................................................................... 21
6. **Operational Overview** ................................................................................................ 23
   - 6.1 Log View ............................................................................................................... 23
   - 6.2 Shell Interaction ..................................................................................................... 24
   - 6.3 OSSEC Example ..................................................................................................... 25
7. **Skedler Licensing** ....................................................................................................... 26
8. **Local DNS** .................................................................................................................. 28
1 OVERVIEW

The SIEMonster V2.5 Free release 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.

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 Format:
Excel

Select Filter:
No Filter

Folder Path:
/home

Upgrade to Premium for more advanced features – sales@siemonster.com
PROVIDED VM IMAGE

The VM image provided is suitable for Virtualbox, VMware Workstation & ESX deployment. The image provides a baseline platform for SIEMonster stack deployment, and is used to for each of the required 5 servers.

The fifth server is the Rancher Server to which the SIEMonster Stack will be joined via secure agent installs.

This image can be downloaded here.

http://releases.siemonster.com/SIEMonster-2.5.ova

This image is used to create all 5 servers.

For production deployment, the minimum recommended RAM requirements for each server are as follows:

<table>
<thead>
<tr>
<th>SIEMonster Server</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makara Rancher Server</td>
<td>8GB</td>
</tr>
<tr>
<td>Capricorn Web Frontend</td>
<td>16GB</td>
</tr>
<tr>
<td>Proteus Ingest Node</td>
<td>16GB</td>
</tr>
<tr>
<td>Kraken Elasticsearch Data Node</td>
<td>16GB</td>
</tr>
<tr>
<td>Tiamat Elasticsearch Data Node</td>
<td>16GB</td>
</tr>
</tbody>
</table>

For development/demo purposes these instances may be deployed with the following RAM options.

<table>
<thead>
<tr>
<th>SIEMonster Server</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Makara Rancher Server</td>
<td>8GB</td>
</tr>
<tr>
<td>Capricorn</td>
<td>8GB</td>
</tr>
<tr>
<td>Proteus</td>
<td>8GB</td>
</tr>
<tr>
<td>Kraken</td>
<td>8GB</td>
</tr>
<tr>
<td>Tiamat</td>
<td>8GB</td>
</tr>
</tbody>
</table>
4 INSTALLATION

4.1 KEY INFORMATION

Image Import

The provided image maybe be imported/opened using Virtualbox or VMWare ESX/Workstation. In Virtualbox click on reinitialize Mac address when importing. Be sure to allocate the following names for reference when importing:

- Makara
- Capricorn
- Proteus
- Kraken
- Tiamat

Network Environment

All 5 instances of SIEMONSTER should be launched within the same network subnet. It is recommended to set a static IP Address for each server

- `sudo nano /etc/network/interfaces` as well as setting the hostname and updating that hostname in the `/etc/hosts` file.
- `sudo nano /etc/hosts`

and change the host name

- `sudo nano /etc/hostname`

Sample hosts file for Makara

```
127.0.0.1  localhost
192.168.15.1  makara
192.168.15.101  kraken
192.168.15.102  tiamat
192.168.15.103  proteus
192.168.15.104  capricorn
192.168.15.105  makara

# The following lines are desirable for IPv6 capable hosts
::1  localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```
Sample interfaces file Makara

```
[Network]
    Name: 2.5.3
    File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.15.105
    gateway 192.168.15.1
    netmask 255.255.255.0
    dns-nameservers 192.168.15.1
```

DNS Settings

DNS settings can be set using a local DNS server. Settings will be an A Record for siemonster.corp.siemonster.local pointing to the IP address of Capricorn, plus a CNAME entry for *.siemonster pointing to siemonster.corp.siemonster.local

Note. The domain name must have 2 subdomain levels, as in the above example.

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

```
/etc/docker/daemon.json
```

```json
{
    "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

**Connectivity**

Ensure connectivity and name resolution using ping to all the monsters by name to ensure the network is setup correctly.

**Username**

The Username/Password for all instances for SSH access is siemonster/siemonster
# PORTS

Recommended firewall rules require the following open port requirements:

- **Internal** = Monster to Monster communications
- **External** = Internet Facing ACL’d/Firewall to specific end points
- **Proteus** = Ingestion Engine of Logs
- **Capricorn** = Viewer of Dashboards
- **Makara** = Orchestrator

<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>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></td>
<td>514, 1516</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Syslog-NG</td>
<td>1514/UDP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1514/TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OSSEC</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/kopf)
- [http://capricorn-ip-address:9200/_plugin/hq](http://capricorn-ip-address:9200/_plugin/hq)
Rancher Network Overview

Incoming Data
Logstash
Syslog-NG
OSSEC

Web Interface

Rancher Server
Container Network
Orchestration

Proteus
Elasticsearch client node

Capricorn
Elasticsearch client node

Kraken
Elasticsearch Master node

Tiamat
Elasticsearch Master node

Makara

SIEMonster
Elasticsearch Single Cluster4 Nodes
Rancher Agents
4.2 RANCHER SERVER

Once all instances have been launched, the first task is to install the Rancher Server. Open an SSH Terminal to the Makara server. In the home folder run the following script:

```
./rancherserver-start.sh
```

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.

4.3 ACCESS CONTROL

Once everything has downloaded, open a web browser and access the Rancher Server on the Makara instance on port 8080

```
http://makara-ip-address:8080
```

Note: If you are running Symantec/Mcafee etc Endpoint protection disable it or allow web socket traffic internal connections or you may experience hosts that won’t add. These product block IPSEC by default which is needed for monster to monster comms. See 502 Gateway DOC for further information under Operations on the website or support website.

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’
Click on ‘Save’

Add the following catalog & click on ‘Save’

<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIEM</td>
<td><a href="https://github.com/siemonster/rancher-free">https://github.com/siemonster/rancher-free</a></td>
</tr>
</tbody>
</table>

Name: SIEM URL: https://github.com/siemonster/rancher-free

4.5 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 name is case sensitive).

proteus =1 for Proteus
kraken = 1 for Kraken
tiamat =1 for Tiamat
4.6 STACK DEPLOYMENT

2 stacks are required for the web interface and should be setup as follows:

Choose the ‘Heaven for SIEMonster app’ from within the SIEM catalog and click ‘View Details’. If nothing is under SIEM hit the refresh button on top right hand corner.

Choose Template 1 and click on ‘Launch’
Return to the SIEM catalog and click ‘View Details’ for the SIEMonster Free App.

**IMPORTANT**

**Under ‘New Stack’, substitute projectname for the required application name. This name will be used for your site domain in the next step. Leave siemonster-project but add a name you would like to call the application in the URL.**

The default is siemonster-project-projectname so your Site Domain name would be projectname.x.x.x where x is whatever you would like to call it i.e. projectname.company.siemonster.local is a possible choice. Just make sure whatever the last value is in the Name field is the first value in the Site Domain field.

Here are some examples:

<table>
<thead>
<tr>
<th>Name</th>
<th>Site Domain Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>siemonster-project-projectname</td>
<td>projectname.company.com.au</td>
</tr>
<tr>
<td>siemonster-project-siem</td>
<td>siem.poc.company.local</td>
</tr>
<tr>
<td>siemonster-project-siemonster</td>
<td>siemonster.corp.client.com</td>
</tr>
<tr>
<td>siemonster-project-siemapp</td>
<td>siemapp.siemonster.com</td>
</tr>
</tbody>
</table>

**Under Configuration Options, Site domain Name delete all 4 values and replace them with a 4 named URL, for example**

**Name:**

*siemonster-project-siemapp* will become

**Site domain name:**

*siemapp.portal.siemonster.com (domain name must have 4 names)*

**Before**

![Image of the configuration options](image-url)
Next set the Elasticsearch JAVA HEAP SIZE per the instance type deployed.

**Note:** If Kraken or Tiamat instances have less than 16GB RAM (i.e., 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 some time to build, depending on the speed of the network connection. This is because the latest SIEMonster Docker image have to be downloaded on the first build. The status can be viewed under Stacks – User.
On completion, the status will turn to green for all items this may take 10 minutes to 8 hrs depending on your internet speed for downloading off the stack containers.

The SIEMonster Web interface will now be available at the configured domain name.  
http://siemonster.corp.siemonster.local  If you are using local dns hosts file for a lab setup go to Chapter 8 and add your host file entries into your machine for this to work.

Default Login: admin@siemonster.com  password: siemonster  
Dashboards for Dockbeat and OSSEC will be fully functional on load with live data.
5 USER SETUP AND MANAGEMENT

Login to the user administration interface at the configured domain name:
http://admin.siemonster.corp.siemonster.local
(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.

**Note:** A user called Admin2 has been created as a backup password of siemonster, change this password and keep it as a spare in case you accidently remove all your admin rights.

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.
6 OPERATIONAL OVERVIEW

6.1 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

<table>
<thead>
<tr>
<th>Started-Once</th>
<th>docker-images-updater</th>
</tr>
</thead>
<tbody>
<tr>
<td>Started-Once</td>
<td>elasticloader</td>
</tr>
<tr>
<td>Active</td>
<td>es-client-1</td>
</tr>
<tr>
<td>Active</td>
<td>es-client-2</td>
</tr>
</tbody>
</table>

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.
6.2 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:
6.3  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
```

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,R 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:[001]:
  Agent information:
  ID:002
  Name:Test-01
  IP Address:172.31.45.51

Confirm adding it(y/n): y
```

Restart OSSEC services on completion.
Enter the Reporting module and set the desired email settings:

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

- **Time zone Settings**
  - America/Chicago

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 until 25 Feb 2017:
Name: Makara Siemonster
Email: siemonster.dec2016@gmail.com
Company Name: SIEMonster
License Key: - This is autogenerated.
8 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
```