LiveAction®

LiveSP

Installation & Operating Guide

LiveAction, Inc.
960 San Antonio Road Ste. 200
Palo Alto, CA 94303, USA
+1 (888) 881-1116
https://www.liveaction.com

20210816-LSPIO_211a

Contents

CHAPTER 1

Introduction

Executive summary

This document describes the installation procedure for LiveSP from the software binaries provided.

Steps for a POC to be successful

1. Schedule a kick-off meeting with the following agenda:
   - define the scope of the POC in terms of number of routers, IWAN architecture, features to demonstrate, etc.
   - share all the documentation describing the POC platform,
   - discuss the limits covered by LiveSP (for example Multi-BR limitations in a PFR environment) and what we recommend for a quick and reliable proof of concept.
2. Provide connectivity to the server on which LiveSP will be installed (SSH connectivity at least).
3. Installation requirements: be sure to follow the recommendation of this document in terms of sizing, OS, partitions, etc.
4. Download binaries and transfer them on the server.
5. Install LiveSP (guidelines hereafter).
6. Provision routers to export flows (see the *CPE Configuration Guide* document).
7. Provision LiveSP fill the AVC and PFR provisioning files, send them to us for checking and feed them into the system.
8. Check that key dashboards work for the few referenced CPEs.

Necessary components

Hardware specifications

For POC, the recommended physical (or virtual) hardware specification for running LiveSP to monitor up to 50 CPEs (demonstration platform) is:

- 4 CPUs
- 16 GB of RAM
- 100 GB of disk (SSD is highly recommended but not mandatory)

Please refer to ***Server specifications for LiveSP platform*** on page 23 and feel free to contact us to get hardware specifications for running LiveSP in a production environment.

OS support

LiveSP provides all mandatory prerequisites and recommended system tools for the following operating systems:

- Amazon Linux 2 64-bit
- Debian 10 Buster 64-bit
- Ubuntu 20.04 server 64-bit (Ubuntu 18.04 is also supported but is not recommended)
- RedHat 8 64-bit (RedHat 7 64-bit is also supported but is not recommended)

If you plan to run LiveSP on a different OS, please refer to ***Mandatory prerequisites*** on page 23 to know the list of mandatory packages which you will need to install and configure manually prior to launch installation.

Support terms regarding Docker

In order to provide the best service available, we have a few notes on our support policy regarding Docker; make sure to check them in ***Docker support terms*** on page 24.

LiveSP installation files

Please make sure you have received the following files from us:

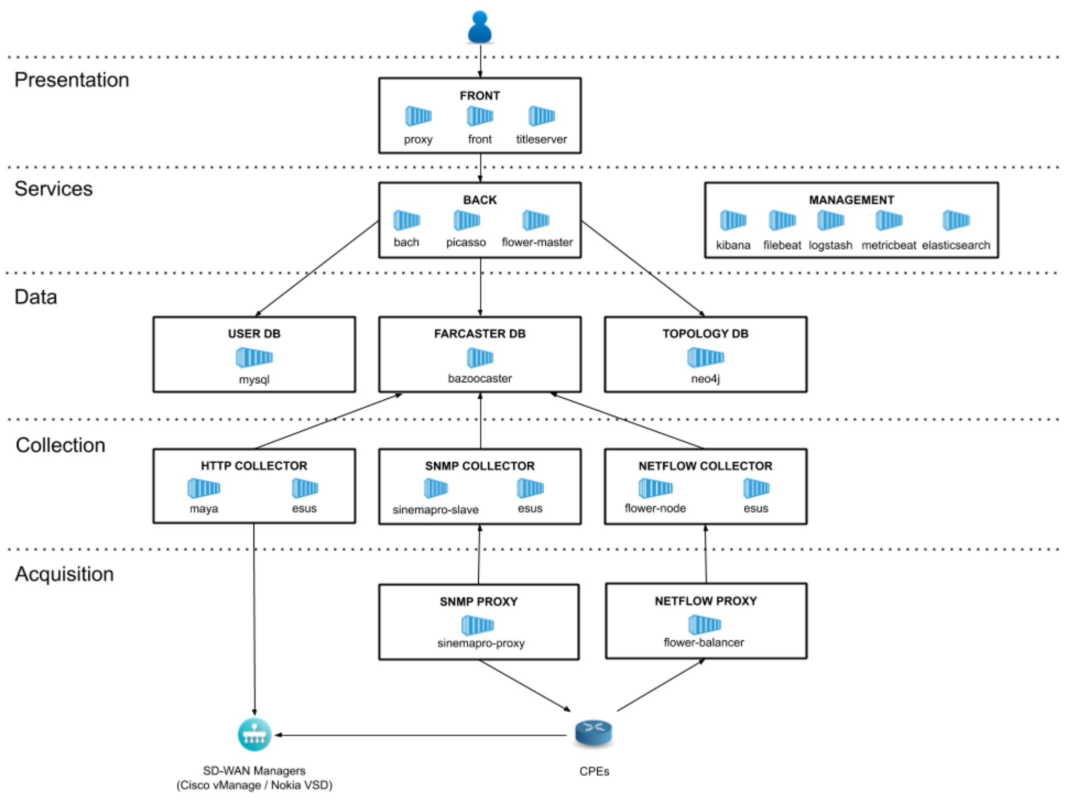
- Binary file
- Custom EZPM profiles to be loaded into the CPEs
- Provisioning templates
- CPE configuration guide
- Provisioning guide
- License key

Important! In case of a multi-server installation, please contact us so we can provide you the correct `hosts` file, which is required in the pre-installation process. However, even without it you will still be able to follow all the pre-installation steps until the software prerequisites deployment section. For a single server installation, you can disregard this note.

LiveSP presentation

Software architecture

As a multi-tenant solution, LiveSP is based on a modular, highly scalable architecture. It is organized into four functional layers. In a single server mode, all layers are installed on a unique server.



Docker components

This is a quick list of different elements in the architecture:

- *Front*: Web JS Application
- *User DB*: Storage for user related data (profiles, dashboards, reports, etc.)
- *Topology DB*: Storage for network topology (CPEs, sites, clients, applications, etc.)
- *Back/User Services*: Back-end micro-services to fetch and set data (reports, dashboards, etc.)
- *Netflow Collector*: Netflow collection agent
- *SNMP Collector*: SNMP collection agent
- *HTTP Collector*: HTTP collection agent
- *SNMP and Netflow Proxies*: Simple proxies in contact with CPEs
- *Management*: In multi-environment, centralizes tools, binaries, logs, etc.

Installing

Pre-installation tasks

Python installation

Our installation process uses a tool called Ansible. It needs to have python installed on all nodes. You can review the exact prerequisites here: ***ansible docs***.

User with sudo privileges

Single server

1. Connect to your server with your root account and create a livesp user account:

   ```
   > useradd -d /home/livesp -m -U -s /bin/bash livesp
   ```

2. If you have not been asked for it, you will have to give this user a new password:

   ```
   > passwd livesp
   ```

3. Give this user the sudo privileges with the following commands:

   ```
   Ubuntu   > usermod -aG sudo livesp
   Debian   > usermod -aG sudo livesp
   RedHat 7 > usermod -aG wheel livesp
   ```

4. For the rest of installation, you must be logged in as this new user:

   ```
   > su - livesp
   ```

Note If you are on Debian, and sudo is not installed on your server, please refer to ***Installation of sudo on Debian 9 and 10*** on page 25.

`umask` must be set to 0022 (and not 0027) for this user.

Multi servers

Repeat this operation on each server part of the architecture and use the same username and password each time.

Network access

Single server

- To enable users to connect onto LiveSP, the ports **80/TCP** and **443/TCP** have to be opened to the presentation layer.
- To be able to send emails, LiveSP (on the back server for multi) has to be able to contact the SMTP server on the appropriate port (for example port **25/TCP** or **587/TCP**).

- If you want access to Neo4J through the web UI (for support team or operation team), ports **7687/TCP** and **443/TCP** have to be opened (on the LiveSP proxy server for multi).

Multi servers

- All servers need to access your management server on:
  - port **5000/TCP**
  - port **9200/TCP** (to centralize logs)
  - port **1202/TCP** (optionally, if you want to install the bundled prerequisites)
- In order to install and update the application and update the application configuration, your management server needs to access all servers on port **22/TCP**.
- In order to allow LiveSP docker services to communicate with each other by the docker swarm system, you need to ensure that following ports are open between all machines that are part of the architecture:
  - port **2376/TCP**
  - port **2377/TCP**
  - port **7946/TCP**
  - port **7946/UDP**
  - port **4789/UDP**

NTP synchronization

Single server

LiveSP collects, stores and reports time-based information: for this information to be reliable, the LiveSP servers and the monitored routers both need to be synchronized by NTP (Network Time Protocol).

To enable NTP, edit the `/etc/ntp.conf` file and in the section Specify one or more NTP servers, add the following line:

```
> server <NTP_IP_ADDRESS> iburst
```

`<NTP_IP_ADDRESS>` must be a valid and accessible NTP server IP address (or DNS name).

Restart the NTP service and wait for a while to get the time synchronized.

```
> sudo service ntp restart
```

To check that the time and date are synchronized, you can run command:

```
> timedatectl
```

You should have “yes” in front of “NTP synchronized”.

Multi servers

Repeat this operation on each server of the architecture.

Binaries upload

Single server

Upload the binary on your server.

Multi servers

Upload the binary on your management machine.

Default installation

For this section, place yourself into the extracted folder of the LiveSP installation binary:

```
> tar xf <BINARY_ARCHIVE>.tar
> cd <BINARY_NAME>
```

The configuration files of the platform will be located in `~/.config/livesp/`, the necessary files for an installation are:

- *`license.key`*: the license file provided by LiveAction
- *`customization.env`*: the application configuration file
- *`hosts`*: the platform architecture description file
- *`docker-compose.yml`*: the orchestration configuration file

License.key

Upload the license file we have provided you in the following location: `~/.config/livesp/license.key`.

Note If you have not received your license file yet, please contact us.

Hosts

This file is used by ansible to install prerequisites and setup a swarm cluster. Servers IPs, users, passwords and swarm labels are described here.

Single server

Copy the default file:

```
> cp documentation/example_files/config/single/hosts ~/.config/livesp/
```

Edit the file with the following values:

- `ansible_ssh_user`: the username of the user created, (if you followed our procedure, it should be “livesp”)
- `ansible_ssh_pass`: the password of the created user
- `ansible_become_pass`: the password of the created user

Multi servers

Copy the file provided by LiveAction into `~/.config/livesp/hosts`.

Docker-compose.yml

This is the docker configuration of LiveSP services. It describes some environment variables, volumes mounted on the host machine, exposed ports, etc.

Single server

Copy the default file:

```
> cp documentation/example_files/config/single/docker-compose.yml ~/.config/livesp/
```

Multi servers

Copy the file provided by LiveAction in `~/.config/livesp/docker-compose.yml`.

Installation launch

If you want to install the applications and all its prerequisites (see the list in ***Mandatory prerequisites*** on page 23), run the following command to launch the installation:

```
> ./scripts/install.sh --install-prerequisites
```

Note If you are installing on a Debian 10 OS, you will need one of the following packages to the launch the installation: gnupg, gnupg2, or gnupg1.

If you have installed the prerequisites by yourself and do not want to install the bundled packages, run the following command to launch the installation:

```
> ./scripts/install.sh
```

The script must end with `failed=0` in the `PLAY RECAP` section.

Advanced configuration

Simple customization parameters

You can add parameters you want to configure in `~/.config/livesp/customization.env`. You will find the list of customizable parameters inside the file `documentation/ANNEXE_customization_reference.txt`.

Furthermore, you will find additional **HowTo** documentations for advanced configuration in the binary folder (for example, if you need to setup Backup and Restore, see HowTo in the corresponding guide).

Extension of the platform through Advanced Services

LiveSP allows for advanced customization to better fit the requirements of your specific platform and offer better services to your enterprise customers tenants.

These being highly sensitive changes, it requires Advanced Services from the LiveAction team: please contact them for more details and exposing your specific needs.

Here are a couple of applications of examples that can be accomplished through this service.

Additional topology types

For example, you may wish to allow groupings of customer sites around multiple SLA levels (Gold, Silver, Bronze), on a topology cluster type called “Site SLA class”.

Additional counters

For example, you may wish to calculate the load on Netflow-monitored interfaces in addition to the default load analysis with SNMP WAN links.

Provisioning mapping columns

For example, you may want to automatically feed additional topology types, or interface bandwidths, according to rules from patterns on interface ifDescription, or SNMP-discovered values.

Update configuration or upgrade version

Update configuration

If at any time you decide to update your current configuration (`customization.env` file, `hosts` file, `docker-compose.yml` file), you will need to run the following (into the binary of your current version) to apply changes:

```
> livesp-stack-update
```

Upgrade version

Once you uploaded the new LiveSP binary for the new release and extracted it, go into the new binary folder and launch following script to upgrade LiveSP version:

```
> ./scripts/install.sh
```

Post-installation and update checks

Once you have completed the LiveSP installation, there are some post checks to be done.

Docker

Logout from livesp account and log in again. Run the following command to check the installation.

```
> docker service ls
```

After 5 minutes, every replica must be “1/1”, meaning that all docker service has been launched and are healthy.

Application

1. Connect to the WebUI and login with account *admin* (default password: *admin*).
2. You will be redirected to the *Monitoring* page. Check that everything is running correctly (✓) except for *collect* and *remoteServer/heartbeat*:
   a. *collect* should be in a critical or warning state since no flow has been collected yet,
   b. *remoteServer/heartbeat* should be in an information state (ℹ) as it is a single server install with no resiliency.
3. If any other service is red, contact us for further assistance.

Preconditions for SNMP and Netflow collection

Network access

SNMP

SNMP proxy server(s) need to access the CPEs on SNMP port **161/UDP**.

Netflow

Your CPEs need to export Netflow data on the Netflow proxy servers port **2055/UDP**.

SDWAN

To enable REST API (to an SDWAN server) collection, the ports **80/TCP** and **443/TCP** have to be opened between the REST API collector and the SDWAN servers.

Connectivity

SNMP

The LiveSP server will use the IP address of the loopback interface to reach the router by SNMP. Check SNMP connectivity between LiveSP server and the router before provisioning is performed by running any `snmpwalk` command from the application to check SNMP connectivity:

```
> snmpwalk -Os -c <COMMUNITY_STRING> -v <SNMP_VERSION> <LOOPBACK_IP> <OID>
```

For example:

```
> snmpwalk -Os -c public -v 2c 172.17.50.1 1.3.6.1.2.1.2.2.1.2
```

Netflow

The IP address associated to the same Loopback interface is configured on CPEs as the source for Netflow monitoring exports to the LiveSP server.

CPE configuration

Please refer the *CPE Configuration Guide* to proceed with CPE configuration (SNMP and Netflow).

Provisioning

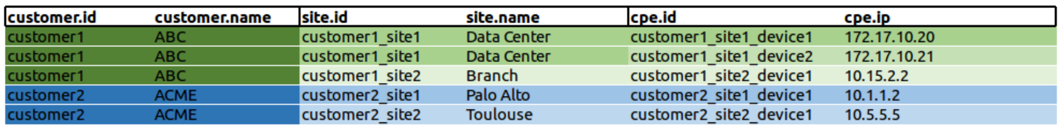
Please refer the *Provisioning Guide* for more details on provisioning and use cases.

Definition

Now that LiveSP has been installed and the CPEs configured and you have successfully completed the post checks, it's time to provision the application. Provisioning is the process of providing LiveSP with all information about the routers to be monitored. Please refer the *Provisioning Guide* for more details on provisioning and use cases.

Using provisioning file

The classic way of provisioning LiveSP is with a CSV file with all needed router information. We maintain different provisioning files (for example AVC and PFR). Templates files provided with the installation binary, here is what a basic seed file can look like:



customer.id	customer.name	site.id	site.name	cpe.id	cpe.ip
customer1	ABC	customer1_site1	Data Center	customer1_site1_device1	172.17.10.20
customer1	ABC	customer1_site1	Data Center	customer1_site1_device2	172.17.10.21
customer1	ABC	customer1_site2	Branch	customer1_site2_device1	10.15.2.2
customer2	ACME	customer2_site1	Palo Alto	customer2_site1_device1	10.1.1.2
customer2	ACME	customer2_site2	Toulouse	customer2_site2_device1	10.5.5.5

Once constructed you must import the file in the web UI. To do this, connect to the LiveSP web UI and login with an admin account; then, in multi-tenant mode, select the *Network* menu, *Provisioning*. Then click on *Manual* at the top right corner: a window to upload the file will open up; select your file and upload it.

Tip For Excel users: Depending on your system region settings, Excel might not use comma character as the default list separator (for example, with France as preferred region settings, Excel considers semi-colon instead): this link gives practical guidance for both Windows and MacOS users.

Using an SD-WAN server

You must give LiveSP the URL of the SD-WAN manager (vManage, VSD, FortiManager, etc.), the credentials of an account that LiveSP will use to connect itself to the server and if needed its associated customer. To do this, connect to the LiveSP web UI and login with an admin account; then, in multi-tenant mode, select the *Network* menu, *Provisioning* and *SD-WAN server library* in the scroll input. Then click on *New* and fill the fields with the correct information and click *Import* after saving to provision LiveSP once.

Check the provisioning

After about 20 minutes (more than 4 flow processing intervals), switch to the monitoring screen: collect entry should be OK.

Check the UI for first graphs. Click on the *Multi-tenant admin* option at the top left corner and then choose the option *Network management*. You will find list of *customers* you have just provisioned.

Click on one of them; if your router configuration is good and if it is sending some flows to the LiveSP server, you will begin to see graphs.

Monitoring

System

The monitoring platform should cover the basic system metrics for all host machines. We recommend a warning severity above 80% threshold on swap memory or disk space at least.

Track configurations update

In order to track any change in application configuration, you can use tool git which is by default installed on management server.

Connect through SSH to the management server and go into the configuration files directory:

```
cd ~/.config/livesp
```

You can then use following command to see all recent changes (date, author, description):

```
git log
```

If you want to check what exactly are the changes for a specific commit, you can use following command:

```
git show <COMMIT_NUMBER>
```

Note Whenever an installation or configuration update is done, all modifications inside configuration files directory are tracked and committed into git.

Track user connection to the application

User connections logs can be found on the management server in the back logs (`<DATA_PATH>/logs/bach.log`).

To easily display all logins since the beginning of the day, you can use the following command:

```
less -R /<DATA_PATH>/logs/bach.log | grep '\[audit\]'
```

You can also search for a specific user:

```
less -R /<DATA_PATH>/logs/bach.log | grep '\[audit\]' | grep <USERNAME>
```

Examples:

```
[INFO] c.l.d.a.d.i.AuthDatabaseManagerImpl {ForkJoinPool-1-worker-6} - [audit] admin
(Admin Admin) logged in
```

```
[INFO] c.l.d.s.i.auth.AuthManagerImpl {reactor-http-epoll-4} - [audit] sso-head-
ers:johndoe logged in via SSO
```

Services

In this section, checks are done either through an SSH CLI command on the management server or through an API call.

We distinguish three importance levels for the services:

- *High*: core function of the product
- *Medium*: important but secondary functions of the product
- *Low*: logging and auto-monitoring system

The documentation on public APIs in LiveSP can be found on:

- the swagger page: `https://<SERVER>/rest-doc/index.html`
- the complementary documentation: `HOWTO_public_rest_api.pdf`

Note These API calls do not use the regular login system but instead the basic authentication implemented for operational (see guide).

Check 1 – LiveSP services are all up and running

Process

- *Bash command*: `livesp-status`
- *API endpoint*: `/api/v1/service/status`
- *Status*: equals “OK” if all services are running else “KO”
- *Severity*: equals “NONE” if all services are running else equals the highest criticality of the services that are not running
- *Messages*: details for each service that is not running (criticality part is detailed in the above paragraph)

Support action

1. Does the service keep failing at startup? What’s the message when it fails?
   - Run command `dksps <SERVICE_NAME>`
2. Check inter-services connectivity
   - Run command `livesp-service-connectivity`
   - If not OK, force restart of all services that failed to be resolved with DNS and see if it solved the issue `dkkill <SERVICE_NAME>`
3. If still not OK, check the last logs of the service, to look for application-level details.
   - Run command `dklogs --tail 50 -f <SERVICE_NAME>`

Example

```
$ livesp-status
{
  "name": "/api/v1/service/status",
  "timestamp": "2020-08-05T14:04:27Z",
  "status": "KO",
  "severity": "MEDIUM",
  "messages": [
    "KO - livesp_dataoperator - Replicas 0/1 - Criticity: MEDIUM"
  ]
}
```

The `livesp_dataoperator` service is not healthy.

```
$ dksps livesp_dataoperator
ID                 NAME                      NODE   CURRENT STATE                    ERROR

impf4uxqmgrdxo75b3d9y28hk   livesp_dataoperator.1  mgt    Running less than a second ago
```

```
g92rsmr0ju6dyevog3lgw4gz0  \_ livesp_dataoperator.1  mgt    Failed 17 seconds
ago "task: non-zero exit (2)"
j63l5awgomjkrum7fxzlc6iz5 \_  livesp_dataoperator.1  mgt   Failed 52 seconds ago
"task: non-zero exit (2)"
0a5ffymddo88ib38drm2yf3xx  \_ livesp_dataoperator.1  mgt   Failed about a minute ago
"task: non-zero exit (2)"
r6ms1owkw4rarafysy6gtuufe  \_ livesp_dataoperator.1  mgt   Failed 2 minutes ago
"task: non-zero exit (2)"
```

The backup and restore management server restart every few seconds (failure to startup correctly).

```
$ dklogs --tail 10 -f livesp_dataoperator
panic: runtime error: invalid memory address or nil pointer dereference
INFO[2020-03-30T19:32:07Z] Server will start on port 8000
INFO[2020-03-30T19:32:07Z] Task purge configured to 30 days.
INFO[2020-03-30T19:32:07Z] Starting server...
ERRO[2020-03-30T19:32:37Z] UpdatePendingTasks: MongoDB find error: connection() : auth
error: sasl conversation error: unable to authenticate using mechanism "SCRAM-SHA-256":
(AuthenticationFailed) Authentication failed.
[signal SIGSEGV: segmentation violation code=0x1 addr=0x28 pc=0x8bf583]
goroutine 8 [running]:
go.mongodb.org/mongo-driver/mongo.(*Cursor).Next(0x0, 0xb51fa0, 0xc000072800, 0x2a)
/root/go/pkg/mod/go.mongodb.org/mongo-driver@v1.1.3/mongo/cursor.go:93 +0x43
liveaction.com/livesp/dataoperator/internal.UpdatePendingTasks()
/root/go/src/liveaction.com/livesp/dataoperator/internal/agentlib.go:87 +0x5c4
main.main.func1()
/root/go/src/liveaction.com/livesp/dataoperator/cmd/data-operator-agent/data_opera-
tor_agent.go:85 +0x20
github.com/robfig/cron/v3.FuncJob.Run(0xa15988)
/root/go/pkg/mod/github.com/robfig/cron/v3@v3.0.0/cron.go:131 +0x25
github.com/robfig/cron/v3.(*Cron).startJob.func1(0xc0000af180, 0xb48e40, 0xa15988)
/root/go/pkg/mod/github.com/robfig/cron/v3@v3.0.0/cron.go:307 +0x69
created by github.com/robfig/cron/v3.(*Cron).startJob
/root/go/pkg/mod/github.com/robfig/cron/v3@v3.0.0/cron.go:305 +0x73
```

The backup and restore management server fail to start because it cannot authenticate while connecting to its database (`livesp_dataoperatordb` service).

Kill the database service with `dkkill livesp_dataoperatordb` and, if problem isn't solved, call L3 support (this service being of medium importance, there is no real emergency but it should be investigated within next few days).

Check 2 – No high or medium service has recently restarted

Process

- *Command*: `livesp-stability`
- *API Endpoint*: `/api/v1/service/stability`
- *Status*: equals "OK" if all services are running since at least 2 days else "KO".
- *Severity*: equals "NONE" if all services are running since at least 2 days else equals the highest criticality of the services that are not running since at least 2 days.
- *Messages*: details for each service that is not running since at least 2 days (criticality part is detailed in the above paragraph).

Support action

1. Check whether a manual operation occurred in last 2 days (if the service was manually killed, a LiveSP installation or upgrade occurred, or the server was rebooted for example, then this is normal).
2. What was the system message when it failed?
   - Run command `dksps <SERVICE_NAME>`
3. Was there a docker heartbeat issue (network communication issue with the manager) on the node of this service? (requires sudo rights)
   - Go to hosting server with `ssh $(getSwarmContainerNodeIP <SERVICE_NAME>)`
   - Run command `sudo journalctl -u docker --since "2 days ago" | grep 'hearbeat'`
4. Check the applicative logs of the service before the service restarted
   - Look for logs in `/data/logs` or in the ELK user interface

Example

```
$ livesp-stability
{
  "name": "/api/v1/service/stability",
  "timestamp": "2020-08-05T14:15:54Z",
  "status": "KO",
  "severity": "HIGH",
  "messages": [
    "KO - livesp_bach - TaskName: livesp_bach.1 - State: Running 29 hours ago - Critic-
ity: HIGH"
  ]
}
```

```
$ dksps livesp_bach
ID                          NAME               NODE     CURRENT STATE  ERROR
q2phpr7ng5fg167435zor6uii  livesp_bach.1      bonite  Running 29 hours ago
q5txlvhq98fggde0uejlwog8d  \_ livesp_bach.1  bonite  Shutdown 29 hours ago
tbvc3mjcul5rmy7h89e45pthi  \_ livesp_bach.1  bonite  Shutdown 2 days ago
z3nu29btw3fwcl2bnsclauv78  \_ livesp_bach.1  bonite  Shutdown 2 days ago
tr8z8whv8ht6mcoa5q4z80zws  \_ livesp_bach.1  bonite  Failed 2 days ago       "task: non-
zero exit (137)"
```

Data Collection

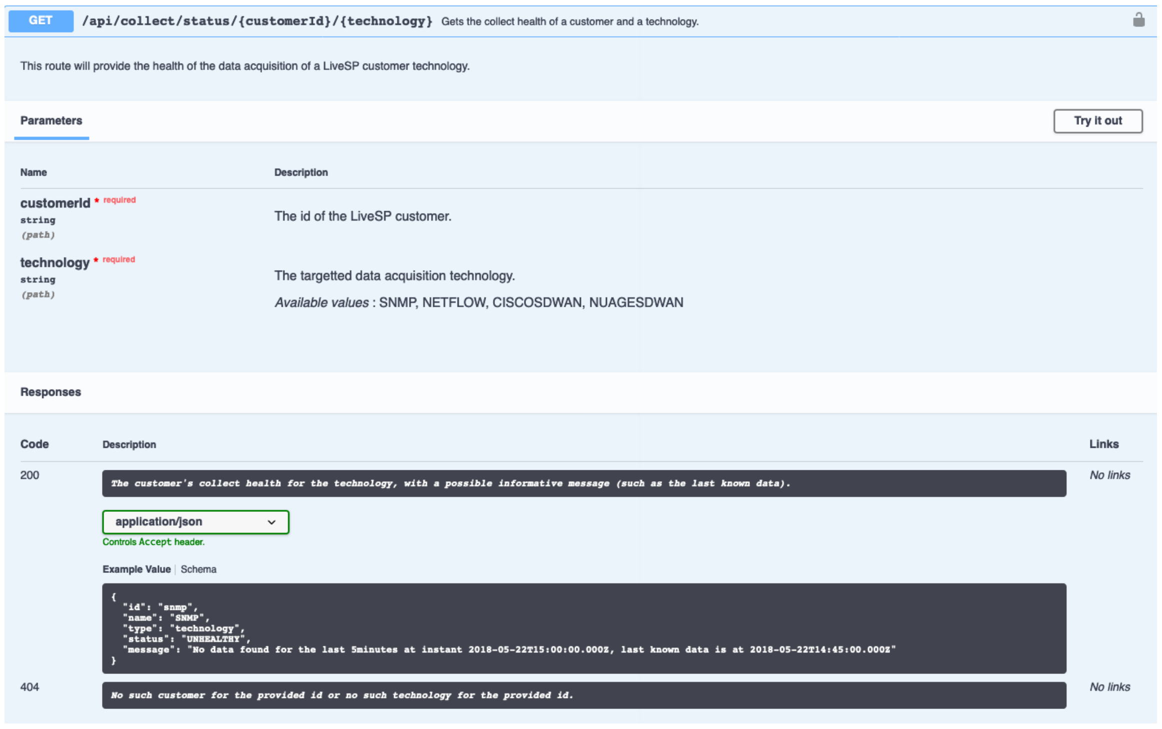
The monitoring for data collection should use the LiveSP public API. The collection API endpoints are meant to track data collection for all clients, from any data source: Netflow, SNMP, and HTTP APIs (such as Cisco vManage, Nokia Nuage, Fortinet CD-WAN, etc.).

The situation for a subset of customers (because they have resigned their contract, or the provisioning was incorrect, or the customer is a single site generating little to no Netflow packets, ...) may distort the collection health diagnostic when we look at the platform on a global scale with API endpoint `/api/v1/collect/status`.

That is why we recommend not using this global API, but rather select a few (1 to 10) reference customers for each collection type, to monitor them in detail, as explained here.

The documentation on public APIs in LiveSP can be found on:

- the swagger page: ***https://<SERVER>/rest-doc/index.html***
- the complementary documentation: *HOWTO_public_rest_api.pdf*



GET `/api/collect/status/{customerId}/{technology}` Gets the collect health of a customer and a technology.

This route will provide the health of the data acquisition of a LiveSP customer technology.

Parameters

Try it out

Name	Description
customerId * required string *(path)*	The id of the LiveSP customer.
technology * required string *(path)*	The targetted data acquisition technology. *Available values* : SNMP, NETFLOW, CISCOSDWAN, NUAGESDWAN

Responses

Code	Description	Links
200	*The customer's collect health for the technology, with a possible informative message (such as the last known data).*	*No links*
404	*No such customer for the provided id or no such technology for the provided id.*	*No links*

application/json

Controls Accept header.

Example Value | Schema

```
{
  "id": "snmp",
  "name": "SNMP",
  "type": "technology",
  "status": "UNHEALTHY",
  "message": "No data found for the last 5minutes at instant 2018-05-22T15:00:00.000Z, last known data is at 2018-05-22T14:45:00.000Z"
}
```

Check 3 – SNMP data collection works fine

Process

- *API endpoint*: `/api/v1/collect/status/<CUSTOMER_ID>/SNMP`
    - *CUSTOMER_ID*: identifier of the customer tenant from provisioning
- *Expected*: `status` value should equal to `HEALTHY`
- *Severity*: High
- *Services* involved in the SNMP collection chain are:
    - livesp_sinemapro-proxy
    - livesp_sinemapro-slave
    - livesp_esus-collector
    - livesp_esus-farcaster*

Support action

1. Are all these services up and running?
    - *Source*: See ***Services*** on page 12
    - *If yes*: Fix this first
2. Was there a recent degradation and when did it start?
    - *Source*: ELK dashboard SNMP Analysis (last 4, 12 and 48 hours)
    - *Expected observation on graphs*:
        - Good stability of the number of realms (total, and per Farcaster node)
        - Relative stability of the number of lines in bridge
        - No alert regarding CPU load, memory load and disk space

- *If yes*: Capture snapshots, summarize observations

3. Is SNMP bridge collection working fine for the customer?
   - *Source:* Run command `checkSNMPCollectedFor <CUSTOMER_ID>`
   - *Expected outputs*: JSON element with a `lineCount` property greater than 0

```
{
  'clientId':'2314459',
  'date':'2020-03-28-01-00',
  'lineCount':7,
  'hostname':'snmpcollector3',
  'ipaddress':'10.156.129.225',
  'bridgePath':'/data/files/bridge/snmp/interface/data-2020-03-28-01-00-00'
}
```

   - *If no (lineCount = 0)*: The source of the problem is on SNMP packet collection side
     - Select 1 or 2 CPEs with wanLinks for this customer (LiveSP inventory screen)
     - Any error or warn in sinemapro-slave logs filtered on this CPE IP address?
   - *If yes (lineCount > 0)*: The source of the problem is on Memdex collection side.
     - Any error in esus-collector logs?

Example

Request URL: ***https://mylivesp.com/api/v1/collect/status/7/SNMP***

Response JSON content:

```
{
  "id": "snmp",
  "name": "SNMP",
  "type": "TECHNOLOGY",
  "status": "HEALTHY",
  "message": "Collect is healthy for technology SNMP"
}
```

Check 4 – Netflow data collection works fine

Process

- *API endpoint*: `/api/v1/collect/status/<CUSTOMER_ID>/NETFLOW`
  - *CUSTOMER_ID*: identifier of the customer tenant from provisioning
- *Expected*: `status` value should equal to `HEALTHY`
- *Severity*: High
- *Services* involved in the Netflow collection chain are:
  - livesp_flower-balancer
  - livesp_flower-node
  - livesp_flower-master
  - livesp_esus-collector
  - livesp_esus-farcaster*

Support action

1. Are all these services up and running?

- *Source*: See ***Services*** on page 12
- *If yes*: Fix this first

2. Was there a recent degradation and when did it start?
   - *Source*: ELK dashboard *Netflow Analysis* (last 4, 12 and 48 hours)
   - *Expected observation on graphs*:
     - Good stability of the number of realms (total, and per Farcaster node)
     - Relative stability of the number of lines in bridge
     - No alert regarding CPU load, memory load and disk space
   - Capture snapshots, summarize observations
3. Is Netflow bridge collection working fine for the customer?
   - *Source*: Run command `checkFlowsCollectedFor <CUSTOMER_ID>`
   - *Expected outputs*: JSON element with a `lineCount` property greater than 0

```
{
  'clientId':'2314459',
  'date':'2020-03-28-01-00',
  'lineCount':10178,
  'hostname':'flowcollector2',
  'ipaddress':'10.154.129.222',
  'bridgePath':'/data/files/bridge/flows/traffic/5minutes/data-2020-03-28-01-00-00'
}
```

   - *If no (lineCount = 0)*: The source of the problem is on Netflow packet collection side
     - Select 1 or 2 CPEs with viewpoints for this customer (LiveSP inventory screen)
     - Any error or warn in flower-node logs filtered on this CPE IP address?
     - Enable debug mode for this CPE: any data after 10 minutes?
     - *If no to all above*: capture Netflow UDP packets with tcpdump on this CPE
   - *If yes (lineCount > 0)*: The source of the problem is on Memdex collection side.
     - Any error in esus-collector logs?

Example

Request URL: ***https://mylivesp.com/api/v1/collect/status/7/NETFLOW***

Response JSON content:

```
{
  "id": "netflow",
  "name": "NETFLOW",
  "type": "TECHNOLOGY",
  "status": "HEALTHY",
  "message": "Collect is healthy for technology NETFLOW"
}
```

Check 5 – REST API collection works fine

Process

- *API endpoint*: `/api/v1/collect/status/<CUSTOMER_ID>/CISCOSDWAN`
  - *CUSTOMER_ID*: identifier of the customer tenant from provisioning
- *Expected*: `status` value should equal to `HEALTHY`

- *Severity*: High
- *Services* involved in the REST API collection chain are:
    - livesp_maya
    - livesp_esus-farcaster*

Support action

1. Are all these services up and running?
    - *Source*: *See* ***Services*** on page 12
    - *If yes*: Fix this first
2. Was there a recent degradation and when did it start?
    - *Source*: ELK dashboard *REST API Analysis* (last 4, 12 and 48 hours)
    - *Expected observation on graphs*:
        - Good stability of the number of realms (total, and per Farcaster node)
        - Relative stability of the number of lines in bridge
        - No alert regarding CPU load, memory load and disk space
    - *If yes*: Capture snapshots, summarize observations
3. Is the API Collector retrieving data for the SDWAN server of this customer?
    - *Source:* Look for response size in applicative logs of Maya

Example

Request URL: ***https://mylivesp.com/api/v1/collect/status/7/CISCOSDWAN***

Response JSON content:

```
{
  "id": "ciscosdwan",
  "name": "CISCOSDWAN",
  "type": "TECHNOLOGY",
  "status": "UNHEALTHY",
  "message": "No data found for the last 5minutes at instant 2020-03-27T08:29:41.372Z,
last known data is at 2020-03-27T06:50:00Z"
}
```

Common Troubleshooting Cases

LiveSP WebUI is not responding anymore

Check docker service

1. Connect via SSH to the server
2. Check that docker service is running with de command:

   ```
   > sudo systemctl status docker
   ```

   The output should display `active (running)` in the “Active” section.
3. If docker service is down, try to restart it with following command:

   ```
   > sudo systemctl restart docker
   ```

If you have still an error while restarting or the service won’t start, contact us.

Check disk usage

Check disk usage with command:

```
> df -h
```

- If your data partition used by LiveSP software is full, contact us to find a solution.
- If other partitions non-related to LiveSP are full (such as system logs), make some space.

Check docker stack

1. Check that docker services replicas are in a “1/1” state (column STATUS) with command:

   ```
   > docker service ls
   ```

2. If none of the containers are running, relaunch the installation:

   ```
   > cd <BINARY_PATH>
   > ./scripts/install.sh
   ```

If you still can’t access after this troubleshoot procedure, contact us to process a deeper investigation.

I need to restart my LiveSP server for some reason

1. Connect with SSH to the server.
2. Stop the containers gracefully:

   ```
   > cd <BINARY_PATH>
   > ./scripts/stop.sh
   ```

3. Check that there isn’t any service left with command:

   ```
   > docker service ls
   ```

4. Reboot your server.

5. Run the application again:

   ```
   > cd <binary_path>
   > ./scripts/install.sh
   ```

Application response time is very slow

Suddenly, without any change in network scope, the application response time is very slow.

1. Connect via SSH to the server.
2. Determine on which partition the data folder is mounted with the command (the output could be for example: `/dev/sda`):

   ```
   > df -h /data
   ```

3. Check the I/O read speed of this last one with the following command:

   ```
   > sudo hdparm -tT <PARTITION_NAME>
   ```

4. If the speed of timing buffered disk reads is under 100 MB/sec, check disk health status.
5. If speed is correct, contact us for a deeper investigation.

The import of my provisioning file fails

1. Confirm that there are no spelling mistakes in provisioning file fields.
2. Check SNMP connectivity between server and routers.
3. Check if the SNMP community is correctly set and configured on your routers.
4. Check and confirm if the license key has been uploaded in the below location (provisioning fails if license key is not uploaded):

   ```
   ~/.config/livesp/license.key
   ```

The import works but there is no data collection

Data is stored every 5 minutes in read-only binary files. If no data appears after 5 minutes:

1. Check if the provisioned interface is the right WAN interface.
2. Check if the interface names are correct and really exist on your routers.
3. For Netflow data:
   a. Check if we effectively receive some data on port 2055:

      ```
      > sudo tcpdump port 2055
      ```

      The output should look like this:

      ```
      tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
      listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
      18:24:45.415258 IP 172.17.50.12.38828 > ubuntu.lo.2055: UDP, length 128
      18:24:46.415282 IP 172.17.50.12.38828 > ubuntu.lo.2055: UDP, length 232
      18:24:49.415568 IP 172.17.50.12.38828 > ubuntu.lo.2055: UDP, length 232
      18:24:50.415432 IP 172.17.50.12.38828 > ubuntu.lo.2055: UDP, length 232
      18:24:53.415579 IP 172.17.50.12.38828 > ubuntu.lo.2055: UDP, length 128
      18:24:55.415701 IP 172.17.50.12.38828 > ubuntu.lo.2055: UDP, length 128
      ```

   b. If you have some entries in your tcpdump output with IP 192.168.14.1, you need to run the following command on your server:

      ```
      > sudo conntrack -F
      ```

   c. Check if they are some data in `/data/files/bridge/flows`.
4. For SNMP data:

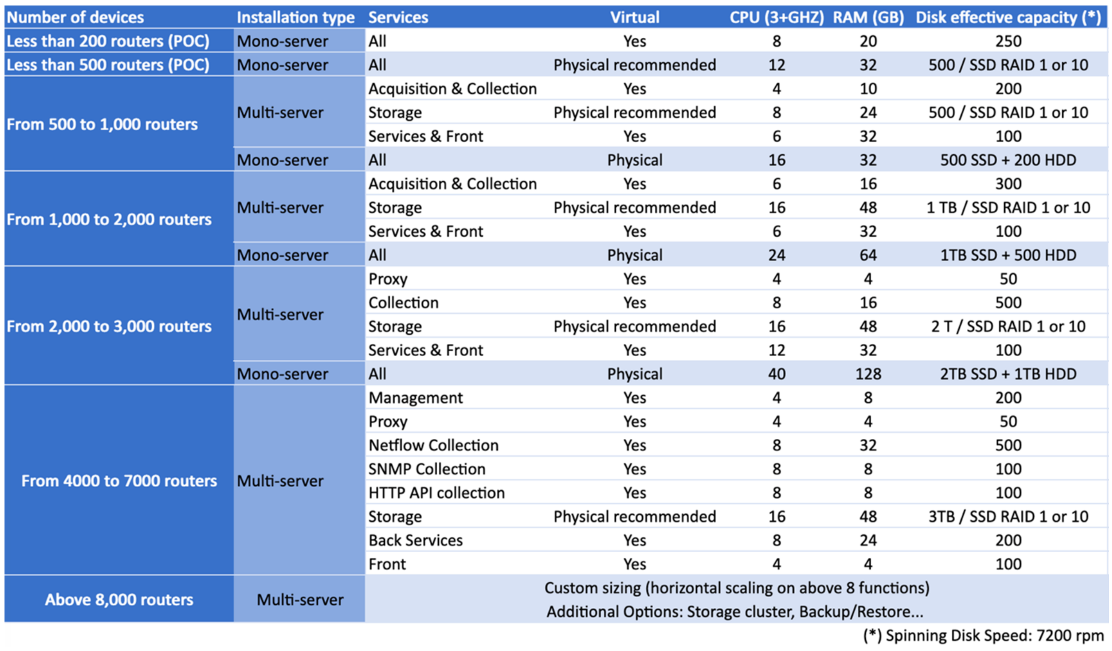
a. Check if we effectively receive some data on port 161.
b. Check if there are some data in `/data/files/bridge/snmp`.

The VM is synchronized by NTP but the date is wrong

Check that the VMware server hosting the VM is also synchronized with NTP.

Appendix

Server specifications for LiveSP platform



Number of devices	Installation type	Services	Virtual	CPU (3+GHZ)	RAM (GB)	Disk effective capacity (*)
Less than 200 routers (POC)	Mono-server	All	Yes	8	20	250
Less than 500 routers (POC)	Mono-server	All	Physical recommended	12	32	500 / SSD RAID 1 or 10
From 500 to 1,000 routers	Multi-server	Acquisition & Collection	Yes	4	10	200
		Storage	Physical recommended	8	24	500 / SSD RAID 1 or 10
		Services & Front	Yes	6	32	100
	Mono-server	All	Physical	16	32	500 SSD + 200 HDD
From 1,000 to 2,000 routers	Multi-server	Acquisition & Collection	Yes	6	16	300
		Storage	Physical recommended	16	48	1 TB / SSD RAID 1 or 10
		Services & Front	Yes	6	32	100
	Mono-server	All	Physical	24	64	1TB SSD + 500 HDD
From 2,000 to 3,000 routers	Multi-server	Proxy	Yes	4	4	50
		Collection	Yes	8	16	500
		Storage	Physical recommended	16	48	2 T / SSD RAID 1 or 10
		Services & Front	Yes	12	32	100
	Mono-server	All	Physical	40	128	2TB SSD + 1TB HDD
From 4000 to 7000 routers	Multi-server	Management	Yes	4	8	200
		Proxy	Yes	4	4	50
		Netflow Collection	Yes	8	32	500
		SNMP Collection	Yes	8	8	100
		HTTP API collection	Yes	8	8	100
		Storage	Physical recommended	16	48	3TB / SSD RAID 1 or 10
		Back Services	Yes	8	24	200
		Front	Yes	4	4	100
Above 8,000 routers	Multi-server	Custom sizing (horizontal scaling on above 8 functions) Additional Options: Storage cluster, Backup/Restore...				

(*) Spinning Disk Speed: 7200 rpm

For hosting the LiveSP main database on a large production platform (above 200 CPEs), please consider the following recommendations:

1. The partition on which the data is stored must be an XFS file system.
2. Very large SPs have installed LiveSP with physical machines for the database and used their Cloud VM infrastructure for the other LiveSP components.
3. Our many years of experience with typical SP cloud infrastructures leads us to strongly recommend physical machines with local SSD over Virtual Machines.

If you still plan to install the database layer on a virtual infrastructure for a large network, we decline responsibility for the end-user quality of experience, as the GUI responsiveness may suffer.

Mandatory prerequisites

If you use a different OS or you want to install them by yourself, please find below mandatory prerequisites list:

- `ansible` (minimum version 2.6) (in a multi-server architecture, it is only needed on your management machine).
- `docker` (minimum version 19.03, see ***Docker configuration*** on page 25)
- `git` (in a multi-server architecture, it is only needed on your management machine),

- `docker-compose` (minimum version 1.14) (in a multi-server architecture, it is only needed on your management machine),
- `jq` (in a multi-server architecture, it is only needed on your management machine),
- `ntp`
- `sudo`
- `bzip2`
- `apache2` (only in multi-server environment)
- `rsync`
- `sshpass`
- `conntrack`
- `crontabs`
- `ethtool`
- `gnupg` (for Debian 10)

Docker support terms

Whenever applicable to the platform, we strongly recommend our clients to buy Docker EE (Basic plan). This note is based on two main reasons:

1. some platforms (such as Red Hat and Oracle Linux among others) are published as being officially supported for the Enterprise Edition,
2. Docker certifies the deployed infrastructure and provides dedicated and extended support for all releases (one year rather than 4 months).

If this is not possible, we will take in charge the installations and upgrades with Docker CE quarterly releases. The default policy is to upgrade to a new quarterly release once a year on production platforms, unless all the following three conditions apply to the currently installed Docker Engine:

- it contains a serious open bug that seriously impedes critical operations,
- it has been upgraded to the latest available hotfix version,
- it gets no more support from the Docker community (more than 4 months old).

In return, we require the following engagements from our clients:

1. the version of installed platforms should be kept within the support life cycle of the distribution, and officially supported by Docker,
2. clients accept to upgrade the Linux kernel version, if a Docker upgrade requires it.

Warnings:

- We can technically provide safe Docker updates and fixes from the Community Edition for the officially supported environments.
- However, for Red Hat and Oracle Linux, please note that we extract Docker CE installation binaries from the CentOS repository; this procedure has not been validated by Docker, and as such, may not be recommended in production environments.

Key links for more information:

- ***Docker release and support strategy***
- ***Docker supported platforms***
- ***Docker Pricing***
- ***Support Life Cycle for Oracle Linux***
- ***Support Life Cycles for other Linux Distributions***

Installation of sudo on Debian 9 and 10

On Debian, `sudo` is not installed by default.

If you have access to the Internet, you can install the package with the following command:

```
> apt-get install sudo
```

You can also install it from our bundled prerequisites (see ***Installation of prerequisites from pre-requisites repository*** on page 26).

Docker configuration

Docker root directory

Please note that your docker root directory (usually `/var/lib/docker`) should be consistent across all servers. The docker root directory is expected to be `/var/lib/docker`. You can change this by editing `~/.config/livesp/customization.env` and adding the following line:

```
DOCKER_ROOT_DIR=/path/to/docker/root/dir
```

If you can't manage to have your docker root directory consistent across all servers, you can create symbolic links to your docker root directory. For example, if you have all your servers configured to have the docker root directory in 1fcvc1fqemgt but on one server it is configured to `/var/lib/docker`, you will add `DOCKER_ROOT_DIR=/data/docker` in your `customization.env` and create a symbolic link on the remaining server `/data/docker` links to `/var/lib/docker`.

Docker manager

Add your user as a Docker manager. Change the username variable with your own Linux username.

```
> sudo usermod -aG docker <USER>
```

Docker engine daemon

To configure docker engine daemon, edit the file `/etc/systemd/system/docker.service.d/docker.conf` (if it does not exist, create folder `docker.service.d` and file `docker.conf` inside it).

```
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd --config-file /etc/docker/daemon.json
ExecStartPost=
ExecStartPost=/usr/sbin/conntrack -F
```

Then edit `/etc/docker/daemon.json` (if it does not exist, create the folders and file).

```
{
  "bip": "192.168.13.1/24",
  "hosts": [
    "unix:///var/run/docker.sock",
    "tcp://0.0.0.0:2376"
  ],
  "insecure-registries": [
    "livesp-registry.liveaction.com:5000"
  ] ,
  "tlscacert": "/etc/docker/certs/ca.pem",
  "tlscert": "/etc/docker/certs/server-cert.pem",
  "tlskey": "/etc/docker/certs/server-key.pem",
  "tlsverify": true
}
```

Important! You have to change the private IP range (192.168.13.1/24) in the `bip` parameter to any other private IP range if you are already using this in your network architecture.

Docker certificates

Manage certificates with LiveSP

You can manage the creation of all the Docker certificates from a script bundled within the LiveSP installer:

```
scripts/generate-docker-certificates.sh
```

This script will ensure you have all the required certificates on your servers and create them if some are missing. You can use the `-f` option to force the generation of all certificates (and CA).

Manage certificates with manually

You can also opt for managing your certificates with your own tools or at hand. In that case please read the following Docker link: ***protect the Docker daemon socket*** and ensure you have the Docker certificates created in the following path:

- `/etc/docker/certs/ca.pem` (CA public key)
- `/etc/docker/certs/server-cert.pem` (Docker daemon certificate signed by the CA)
- `/etc/docker/certs/server-key.pem` (Docker daemon certificate private key)
- `/home/<USER>/.docker/ca.pem` (CA public key)
- `/home/<USER>/.docker/cert.pem` (Docker daemon certificate signed by the CA)
- `/home/<USER>/.docker/key.pem` (Docker daemon certificate private key)

Restart docker service

```
> sudo systemctl daemon-reload
> sudo systemctl restart docker
```

Installation of prerequisites from prerequisites repository

Prerequisites can be installed in different ways:

While installing or updating the application

This operation will install all needed prerequisites and update your packages if they don't meet the minimum required version.

```
> cd <BINARY_PATH>
> scripts/install.sh --install-prerequisites
```

If you want to also update your packages to the latest release of our bundled prerequisites you can run the following command:

```
> cd <BINARY_PATH>
> scripts/install.sh --install-prerequisites --update-strategy latest
```

Independently of the installation

To install all the prerequisites or only one on single server or multi-servers. You can use the following command (will show the documentation).

```
> cd <BINARY_PATH>
> scripts/install_prerequisites.sh --help
```

Examples

- Install all prerequisites

```
> cd <BINARY_PATH>
> scripts/install_prerequisites.sh
```

- Install a single server with prerequisites latest version available on our bundled repository

```
> cd <BINARY_PATH>
> scripts/install_prerequisites.sh --update-strategy latest
```