**Introduction**

This document contains a detailed description of the symbIoTe Smart Space docker deployment process. As a result of the following steps, you will setup and run the docker stack of symbIoTe Smart Space interfacing to IOT platforms or L3/L4 Smart Devices. You will also register your symbIoTe Smart Space to the symbIoTe Core services offered by SHAPES project, which collects the metadata for all symbIoTe-enabled Smart Spaces. You will also be able to register and access resources in Smart Spaces.

To run the symbIoTe Smart Space you need to have public IP address.

**1 Preparation steps**

* 1. **Register user and configure symbIoTe Smart Space to symbIoTe Core Services.**

The first step is to create a service owner user in the symbIoTe Core Admin web page. During registration, you have to provide:

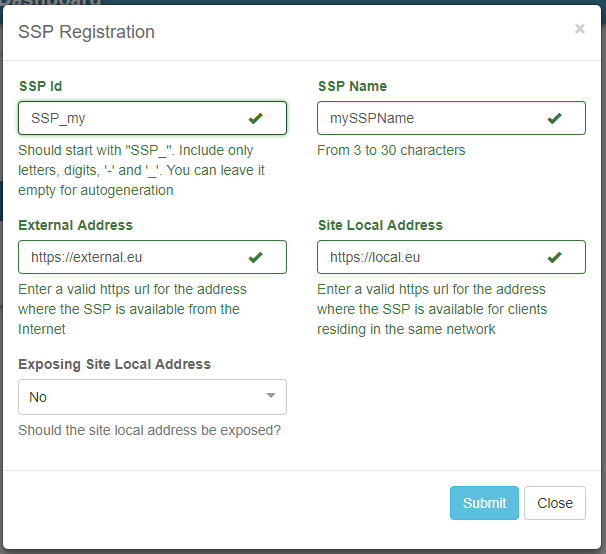
* username
* password
* email
* user role (Service Owner in this case)



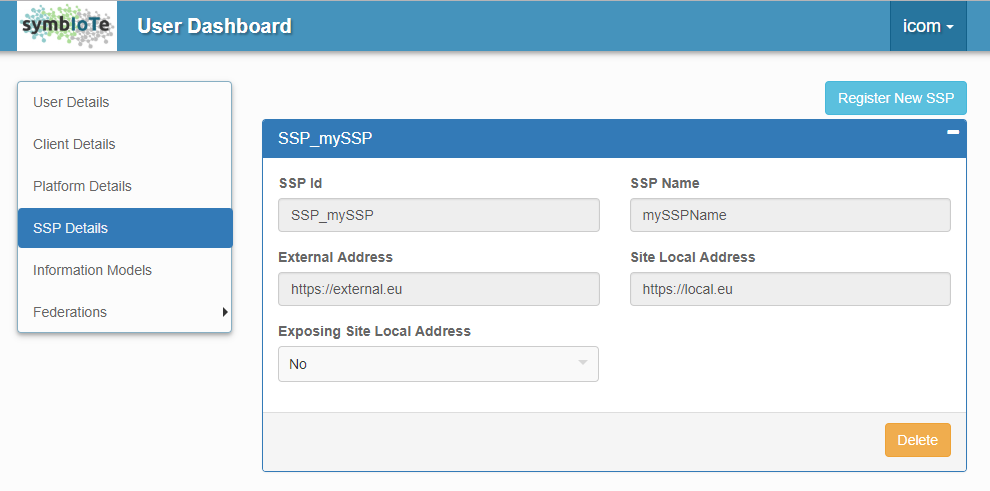
Afterwards, you can log in as the new user and register your symbIoTe Smart Space (SSP). To this end, you have to click on the ***SSP Details*** panel and then on ***Register New SSP*** button on the upper right corner.

Then, you have to provide the following details:

* Preferable SSP id.
* SSP Name.
* External Address: a valid https URL for the address where the SSP is available from the Internet.
* Site Local Address: a valid https URL for the address where the SSP is available for clients residing in the same network.
* Choose if the site local address should be exposed.



In this step your symbIoTe Smart Space (SSP) is registered in the symbIoTe Core Services. You will see the panel of the newly registered symbIoTe Smart Space (SSP) and check its details by clicking on its header.



Finally, you can delete the registered SSP by clicking the delete button on the bottom right corner of the SSP details.

* 1. **Install prerequisites in the virtual machine.**

In the virtual machine install the following:

* docker
* docker-compose

The virtual machine running the docker stack of Symbiote Smart Space must expose the HTTPS port 443 to internet with a public IP address. Also, the virtual machine must have the ability to access the symbIoTe Core Services through port 443.

* 1. **Create the folder and the necessary subfolders for your deployment.**

In virtual machine create the **SymbioteSmartSpace** folder:

mkdir SymbioteSmartSpace.

Inside **SymbioteSmartSpace** folder create the **SmartSpaceMiddleware** folder and copy the **application.properties** file.

Update the **application.properties** file with the correct configuration:

* Set to ***ssp.id*** the value assigned to **SSP Id** during the SSP registration phase (step 1.1).
* Set to ***ssp.url*** the public IP URL of your virtual machine.
* Update the ***symbIoTe.core.interface.url*** , ***symbIoTe.cloud.interface.url*** and ***symbIoTe.rap.cram.url*** with the correct Core Services URL.

Inside **SymbioteSmartSpace** folder create the **AuthenticationAuthorizationManager** folder and copy the **bootstrap.properties** and **cert.properties** file.

Update the **bootstrap.properties** file with the correct configuration:

* Update the ***symbIoTe.core.interface.url*** with the correct Core Services URL.
* Update the ***symbIoTe.interworking.interface.url*** with the public IP of your virtual machine.

Update the **cert.properties** file with the correct configuration:

* Set to ***serviceId*** the value assigned to **SSP Id** during the SSP registration phase (step 1.1).
* Set to ***serviceOwnerUsername*** and ***serviceOwnerPassword*** the credentials of the user registered through administration web page to symbIoTe Core services.
* Update the ***coreAAMAddress*** with the URL of Core Services AAM.
  1. **Obtain certificate files.**

To secure communication of symbIoTe Smart Space, an SSL certificate is needed. Obtain certificate files **fullchain.pem** and **privkey.pem** as described in:

https://github.com/symbiote-h2020/SymbioteCloud/wiki/2.1-Configuration-of-NGINX#2111-obtaining-the-ssl-certificate

Inside **SymbioteSmartSpace** working folder create the **nginx-certificates** folder.

Copy to nginx-certificates folder the certificate files **fullchain.pem** and **privkey.pem**.

$ sudo cp /etc/letsencrypt/live/{your domain}/fullchain.pem nginx-certificates/

$ sudo cp /etc/letsencrypt/live/{your domain}/privkey.pem nginx-certificates/

**1.5  Configuring NGINX with HTTPS.**

Nginx microservice is used for redirecting requests to Symbiote Smart Space Docker microservices from IOT platforms, Smart Devices, Symbiote enabled applications. Nginx needs to be configured so that it redirects correctly to appropriate microservice endpoints. Copy the Nginx configuration file nginx-prod.conf inside **SymbioteSmartSpace** folder.

* 1. **Configuration of docker compose files.**

Inside **SymbioteSmartSpace** folder copy the **docker-compose-prod-swarm-L3.yml** and **docker-compose-swarm-L3.yml** files.

* **docker-compose-prod-swarm-L3.yml** is a compose file to deploy the nginx microservice.
* **docker-compose-swarm-L3.yml** is a compose file to deploy the **AuthenticationAuthorizationManager**, **SmartSpaceMiddleware** and **MongoDB** microservices. In case the virtual machine runs behind a proxy, update the proxy settings according to your proxy server. If no proxy is used, then comment proxy settings parts by putting # in the beginning of line. Parameters related to proxy are: JAVA\_HTTP\_PROXY, JAVA\_HTTPS\_PROXY, JAVA\_SOCKS\_PROXY and JAVA\_NON\_PROXY\_HOSTS.
  1. **Run the docker stack.**

To deploy the Symbiote Smart Space docker stack run:

* docker swarm init
* docker stack deploy -c docker-compose-swarm-L3.yml -c docker-compose-prod-swarm-L3.yml *nameOfYourStack*

Notice that *nameOfYourStack* is the nameof the deployed docker stack.

Check if the docker stack is up and running. In the command line execute:

docker ps to list the microservices of the docker stack.

To remove the stack execute the following command:

docker stack rm *nameOfYourStack* to remove the docker stack.

* 1. **First tests to check the state of the docker stack.**

From a browser run the commands :

* <https://yourdomain/innkeeper/public_resources>

The correct response at the beginning is an empty json array [] .

* <https://yourdomain/aam/get_available_aams>

Returns from Symbiote Core Services the information about all available AAMs in the system.