Server installation instructions

Xesar 3.0

Installation instructions for servers with Ubuntu 16.04/18.04

KPM.X.Xesar 3.0 Installationsanleitung Ubuntu 16.04_18.04 EN | 07/2019 | Subject to misprints and technical changes.

Contents

1 Preface 2
2 Requirements 3
3 Install Ubuntu (valid for 16.04 and 18.04) 3
4 Create Docker Machine 8
5 Install Xesar 3.0 10
1. Install Installation Manager 10
2. Add Xesar system 11
6 Data backup 12

1 Preface

These instructions show the installation of the Xesar 3.0 locking system software on a server with the operating system Ubuntu 16.04 or 18.04. The creation of the necessary IT and server environment is not part of these installation instructions and has to be provided by the customer. EVVA does not bear any responsibility.

Before the installation, check and confirm that the Xesar 3.0 system requirements are met, as required by the project checklist and the system manual.

You can download the current project checklist from the Xesar homepage https://www.evva.com/at-de/produkte/elektronische-schliesssysteme-zutrittskontrolle/xesar/.

We strongly recommend you to only carry out the Xesar 3 installation in close cooperation with the customer's responsible IT administrator.

Instructions for installing Xesar 3.0 on Linux Ubuntu 16.04 or 18.04 LTS server.

2 Requirements

- **Admin Client PC WIN 10 PRO** with installed docker and installation manager
- **Server** with VM Ubuntu 1604 or 1804
- **Xesar 3.0 installation requirements are fulfilled**

3 Install Ubuntu (valid for 16.04 and 18.04)

1. Download Ubuntu 18.04:

   releases.ubuntu.com/18.04/ubuntu-18.04.1.0-live-server-amd64.iso

   Tutorial for Ubuntu Installation:

   https://tutorials.ubuntu.com/tutorial/tutorial-install-ubuntu-server#0

   bootable USB stick:
   https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-windows#0

   Follow the installation steps.

2. After the successful installation of Ubuntu select the option ***open ssh server***. If this option is not available, use the command

   sudo apt install openssh-server

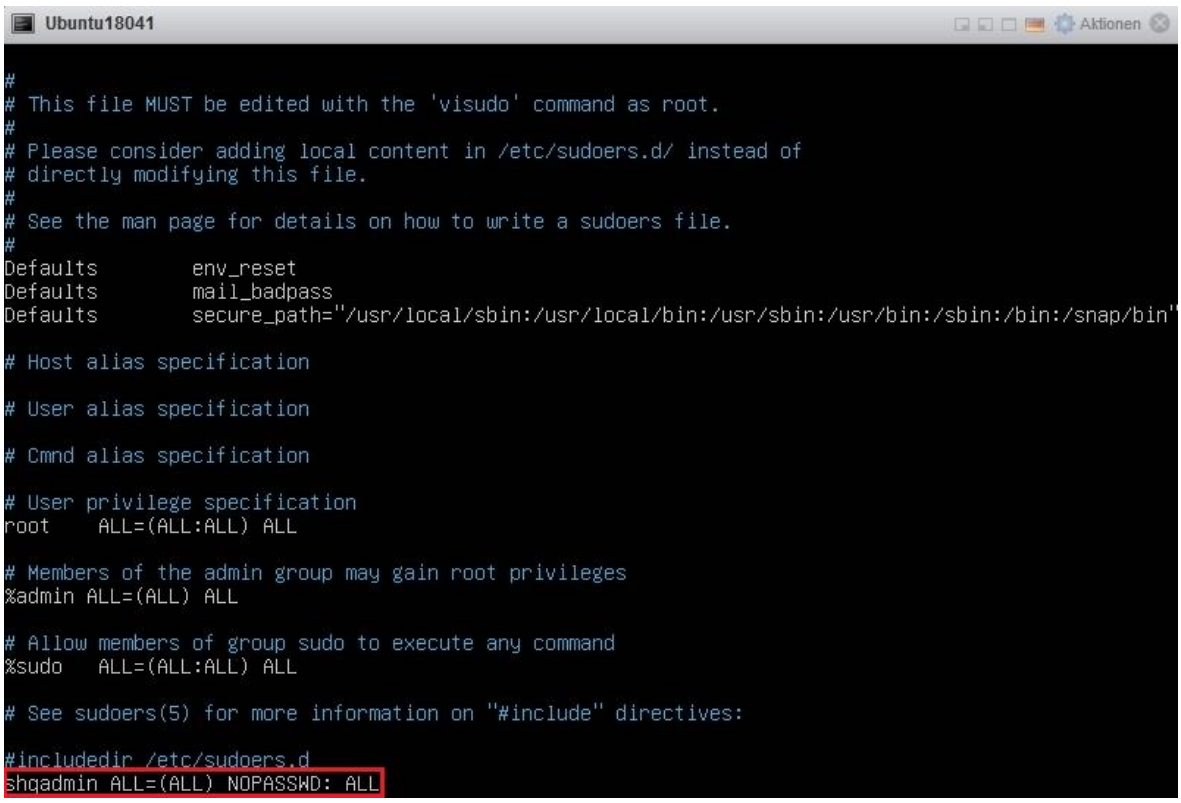
   it can be retrospectively installed via the Linux console. If "sudo without password" (see point 2 below) has not yet been configured, you will be requested to enter the *user* password.

3. To set up sudo without a password, enter the following commands in the Linux console:

   sudo visudo

4. (A password will be requested and the file /sudoers.d will be opened)
5. Then scroll to the end of the editor and type the following commands under the last line:

username ALL=(ALL) NOPASSWD: ALL

6. 

7. Save with CTRL+O and close with CTRL+X (description in the editor)
8. Thereafter the command

   sudo visudo

   functions without a password prompt.

9. Create *a ssh keypair* (default is rsa encryption) with the following command in the Linux console:

   ssh-keygen or ***ssh-keygen -t rsa -b 4096***

10.

```
shqadmin@shqserverubuntu18041:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/shqadmin/.ssh/id_rsa):
Created directory '/home/shqadmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/shqadmin/.ssh/id_rsa.
Your public key has been saved in /home/shqadmin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:S/mcKAmWOMqcTx+AodNdRBifmZDe/xxQq+KNeSehHic shqadmin@shqserverubuntu18041
The key's randomart image is:
+---[RSA 2048]----+
|   . o*+         |
| o =o+ +  .      |
|o +.+.=  . .     |
| + +.o. ...      |
|  = = ..So       |
|   + o.++=..     |
|    ..E*+++.     |
|      ==+ +      |
|     ... o       |
+----[SHA256]-----+
```

The ssh key is stored by default at /home/**user**/.ssh on the Linux server.

11. In our example User = **shqadmin**, which we created during the Linux installation.

12. The created public key (.pub) which belongs to the keypair, must be added to the authorised keys on the Linux server via the Linux console. To do this, switch to the previously created directory with the first command line and add the key as the second line:

cd /home/user/.ssh

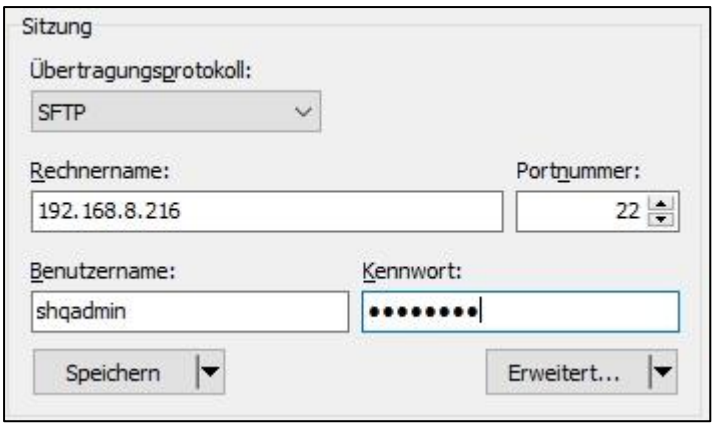
cat id_rsa.pub > authorised_keys

13.

```
shqadmin@shqserverubuntu18041:~$ cd /home/shqadmin/.ssh
shqadmin@shqserverubuntu18041:~/.ssh$ cat id_rsa.pub > authorized_keys
```

14. Now install a program, e.g. putty or WINSCP, to transfer data securely from the client (physical Win10PRO PC) to the server and vice versa). In this example WINSCP is used. (a freeware program: https://winscp.net/eng/download.php)

15. Login to the server via winscp:

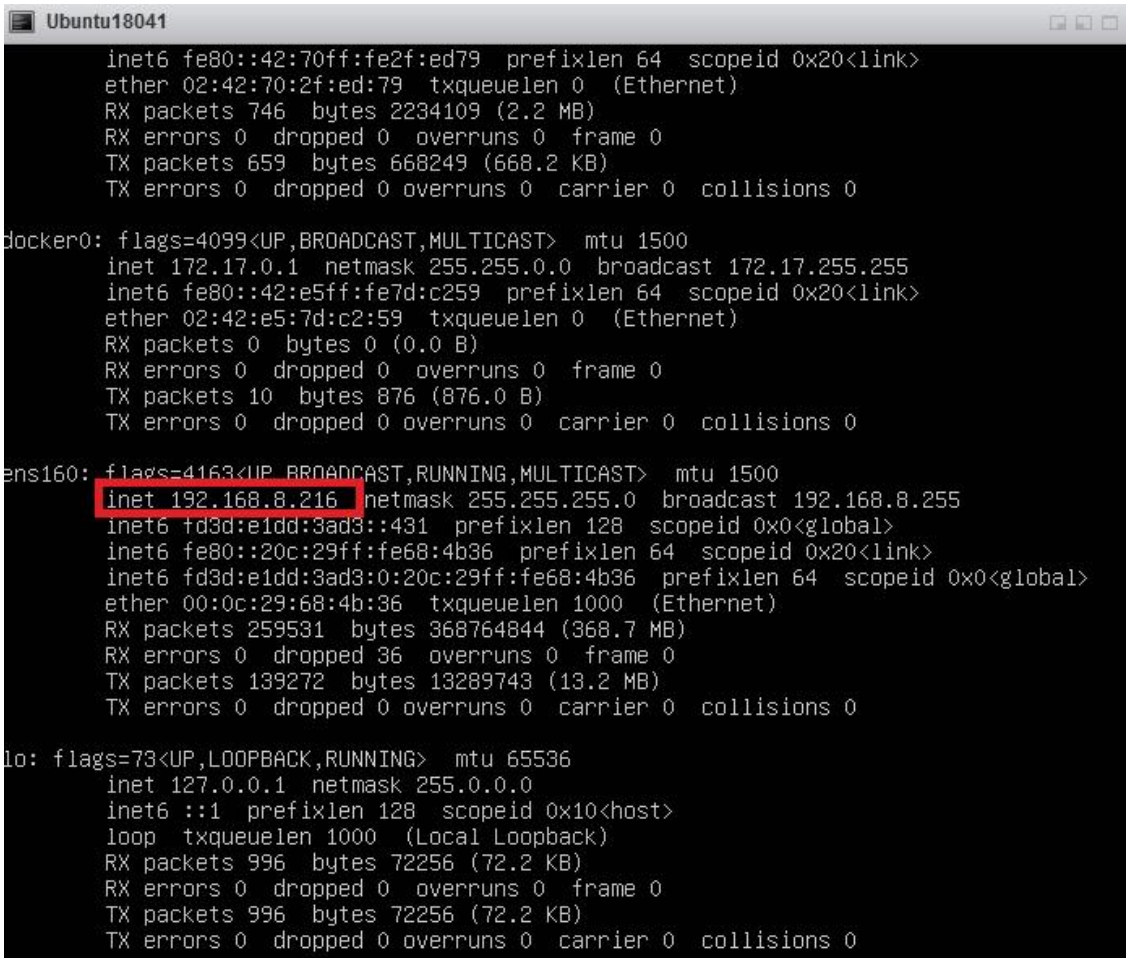


Transfer protocol SFTP
Host name = IP address of the servers (can be determined in the Linux console with the command

ifconfig

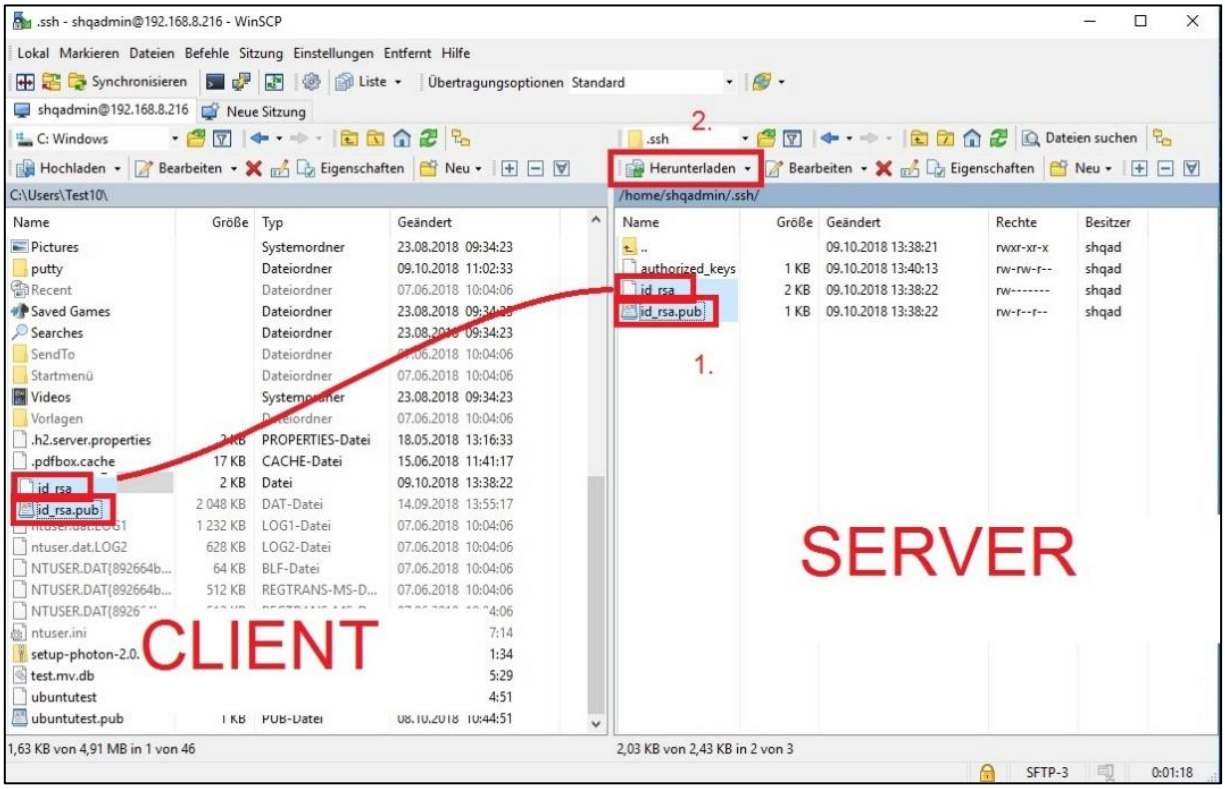
)

16.



Port = standard 22
User and password correspond to the user and the user's password as on the Linux server

17. Now copy the private ***key id_rsa*** and ***public key id_rsa.pub*** via WINSCP to the client (in our example from */home/shqadmin/.ssh* on the server to *C:/Users/Test10* on our WIN10 client (Test10 is the name of the User on our physical WIN10 PC!)



18. Open the windows console (with CMD in search, execute as Admin with right mouse)
Switch to the directory in which the private key id_rsa was stored with the following command in the Windows console.

cd *C:/Users/Test10*

19. (this may differ depending on specification)

```
C:\WINDOWS\system32>cd C:/Users/Test10
```

4 **Create** Docker Machine

Enter the command to create the *docker machine* in the windows console (also from the directory where the public key is located)

```
C:\Users\Test10>docker-machine create --driver generic --generic-ip-address=192.168.8.216 --generic-ssh-key=id_rsa --generic-ssh-user=shqadmin xs3ubuntu1804
Running pre-create checks...
Creating machine...
(xs3ubuntu1804) Importing SSH key...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with ubuntu(systemd)...
Installing Docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env xs3ubuntu1804
```

The general command is:

docker-machine create --driver generic --generic-ip-address (IP Server address) --generic-ssh-key (name of the public keys) --generic-ssh-user (name of the user for whom the Ubuntu Server was created) (name of the docker machine)

Explanation:

command part	explanation
docker-machine create	is the general command to create a Docker Machine
--driver generic	the generic driver for installing Docker on the server
--generic-ip-address	the IP address of the server
--generic-ssh-key	the detail of the public keys used (are executed from the directory in which stored, otherwise the complete path must be entered)
--generic-ssh-user	Specification of the ssh user (in our case "shqadmin") then after one space, the name of the Docker Machine follows (in our case xs3ubuntu1804)

<u>Note:</u> The complete process *docker machine create* takes 2-10 minutes, depending on the computer

In case of an unexpected error message, you can cancel the process by exiting the Windows console. Then open the Windows console again and delete the incomplete docker machine

docker-machine rm "the given name"

e.g. docker-machine rm xs3ubuntu1804

Afterwards you can repeat the command under **point 9** with the addition --*debug,* to get an exact error message:

docker-machine --debug create --driver generic --generic-ip-address (IP address of the server) --generic-ssh-key (name of the public key) --generic-ssh-user (name of the user for whom the Ubuntu Server is created) (name of the docker machine)

If an error message relates to the **ssh connection**, please check the *user* again with sudo without password or check the storing of the **ssh-key**.

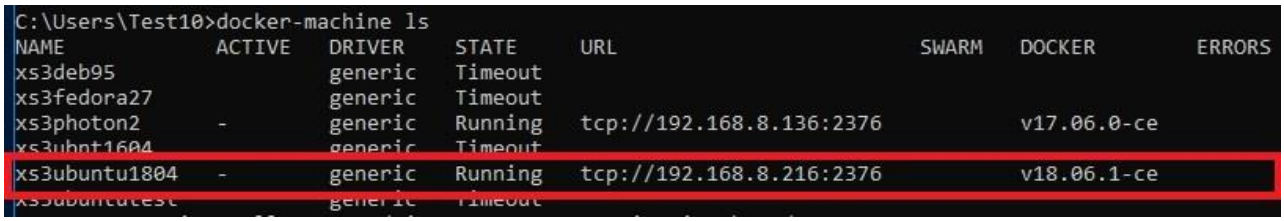
If an error message relates to **docker** (e.g. *sudo get docker version not found* or similar), try to install Docker manually in the Linux console with the following command:

sudo apt install docker.io

After successfully creating the Docker Machine, check in the Windows console with the command:

docker-machine ls

whether the docker machine really runs.



5 Install Xesar 3.0

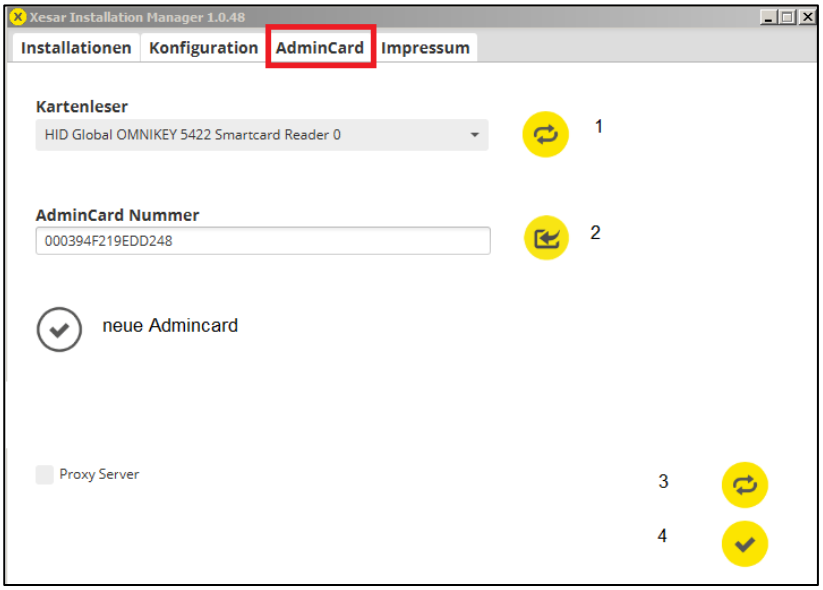
1. Install Installation Manager

In case the Installation Manager is not yet available, you can download it using the following link:

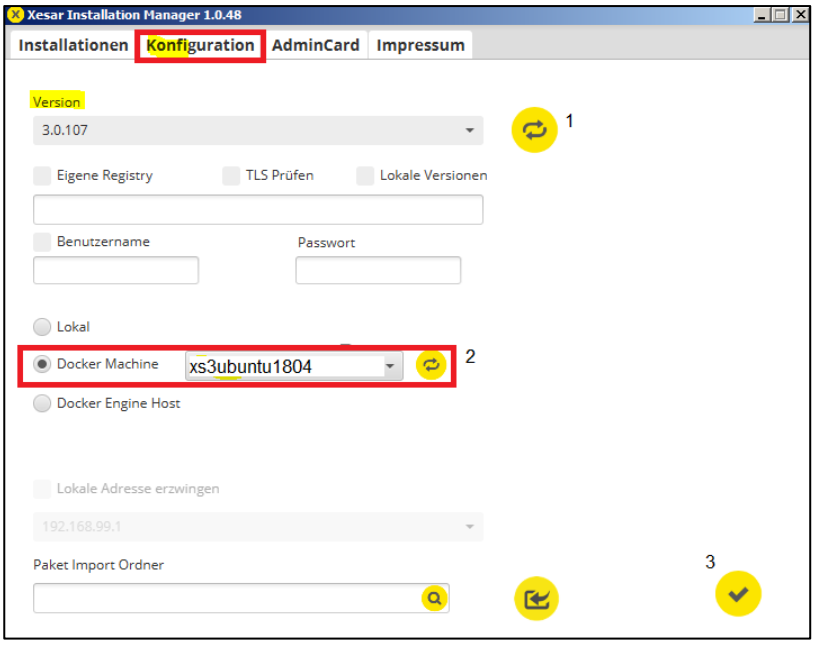
https://www.evva.com/at-de/produkte/elektronische-schliesssysteme-zutrittskontrolle/xesar/xesar-software-download/

Plug in the coding station and start the Installation Manager.

Select the *AdminCard* tab, then the required card reader (1) and enter the AdminCard number by clicking on the button (2).

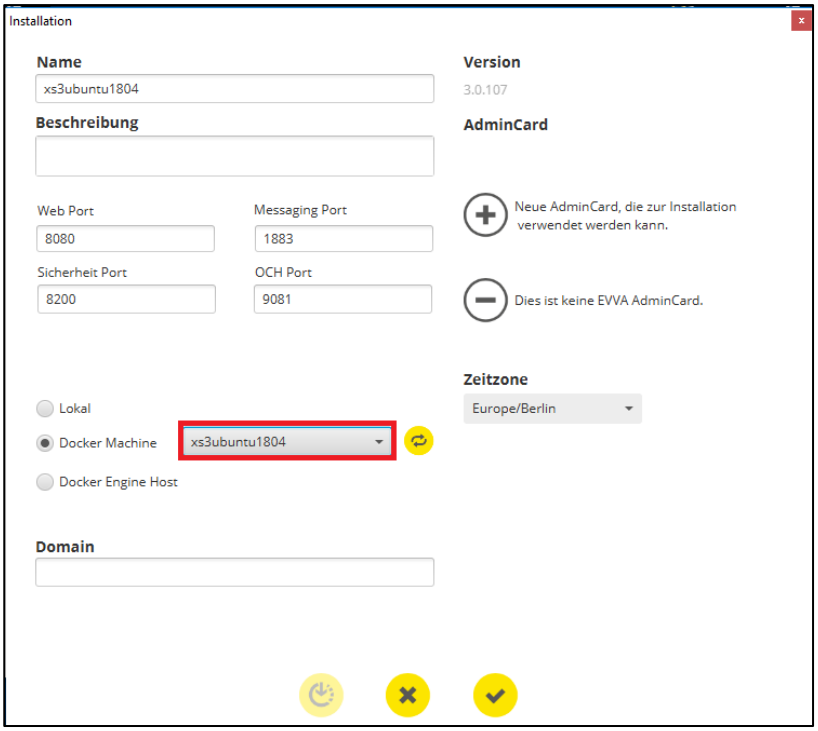


Switch to the ***Configuration*** tab and select the *Docker Machine*



2. Add Xesar system

Open the ***Installation*** tab and add a new installation with "+". After assigning a name and port, select the ***Docker Machine***.



Note:
To update Xesar 2.2, please enter the database path for the import.

After creating the system, you can start and commission the system according to the system manual.

6 Data backup

The following data must be saved:

- **Admin PC** (Windows 10 PRO physical PC). [XesarUser] is a placeholder for the Windows User for whom the Xesar3 installation was performed, e.g. admin, etc.

  - C:System\Users\[XesarUser]\.xesar-1.0.XX\system name
  - C:System\Users\[XesarUser]\.xesar-cs
  - C:System\Users\[XesarUser]\.docker
  - ssh key

IMPORTANT:
From the version Xesar 3.0 SP1, a manual and automatic backup can be performed in the Installation Manager.

- **VM Server** back up the following:

  - Take a snapshot of the VM after each major or important change
  - Generally a mirroring of the whole partition, better still mirror the whole hard disk on which the Xesar VM (for example Ubuntu) is installed - as is usual with servers
  - ssh key

- **Physical server** back up the complete hard disk.