SIEMENS
Ingenuity for life

Creating and managing Edge apps using the example of the "SendMail" app

SIMATIC WinCC Unified (Engineering) V16
SIMATIC HMI Unified Comfort Panels
Device-managed EDGE

https://support.industry.siemens.com/cs/ww/en/view/109778780

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG (“Siemens”). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens’ products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens’ products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer’s exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: https://www.siemens.com/industrialsecurity.

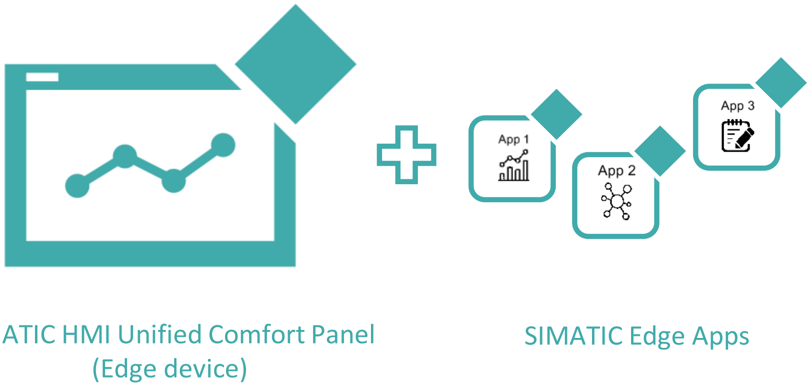
Table of contents

1 Introduction

1.1 Overview

The SIMATIC HMI Unified Comfort Panels are part of the newly developed SIMATIC WinCC Unified system. In addition to many new functions offered by the new hardware and software, the SIMATIC HMI Unified Comfort Panel can also be used as an edge device.

Figure 1-1



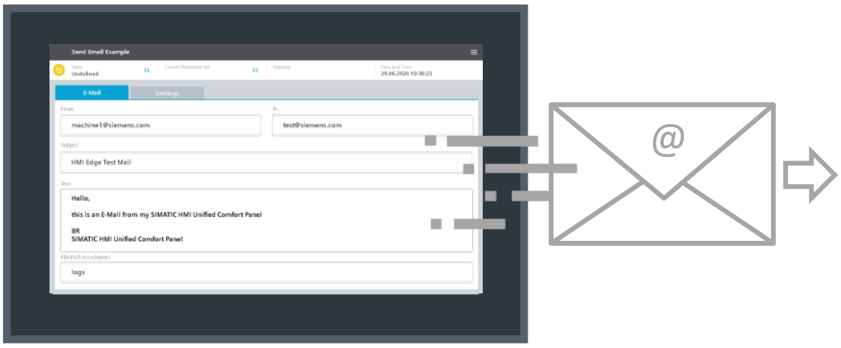
With the help of this additional option, the SIMATIC WinCC Unified system can be extended to carry out other functions and meet customer-specific requirements.

1.2 Principle of operation

This application example will describe in principle how you can program your own edge app and which steps are necessary in order to access data from the runtime.

In addition, this application example contains the code and the TIA project for the SendMail app. You can configure all email settings, then send them via email, in the runtime of the SIMATIC HMI Unified Comfort Panel.

Figure 1-2



From: machine1@siemens.com

To: test@siemens.com

Subject: HMI Edge Test Mail

Text:

Hello,

this is an E-Mail from my SIMATIC HMI Unified Comfort Panel

BR
SIMATIC HMI Unified Comfort Panel

File Path Attachment: logs

Note

> SIMATIC Edge requires a license to be used in a production environment. You do not need a license for the app development and testing phase.
>
> https://www.siemens.com/comfort-edge-activation

1.3 Required knowledge

The following knowledge is required for this application example:

- Programming knowledge in Node.js
- Knowledge working with Docker and Docker containers

1.4 Components used

The following hardware and software components were used to create this application example:

Table 1-1

Component	Quantity	Item number	Note
SIMATIC WinCC Unified V16 (Engineering) Update 1	1	6AV2153-....1-6...	Engineering in TIA Portal.
SIMATIC HMI Unified Comfort Panel MTP1200	1	6AV2128-3MB06-0AX0	Alternatively, any other SIMATIC HMI Unified Comfort Panel can be used.
Docker Desktop 2.3.0.2	1	-	https://hub.docker.com/editions/community/docker-ce-desktop-windows/
Visual Studio Code	1	-	https://code.visualstudio.com/ Any other editor can be used as an alternative.

This application example consists of the following components:

Table 1-2

Component	File name
Documentation on the application	109778780_SendMail_EdgeApp_V10_ en.pdf
TIA project	109778780_SendMail_PROJ_V10.zip
SendMail App Source Code	109778780_SendMail_CODE.zip

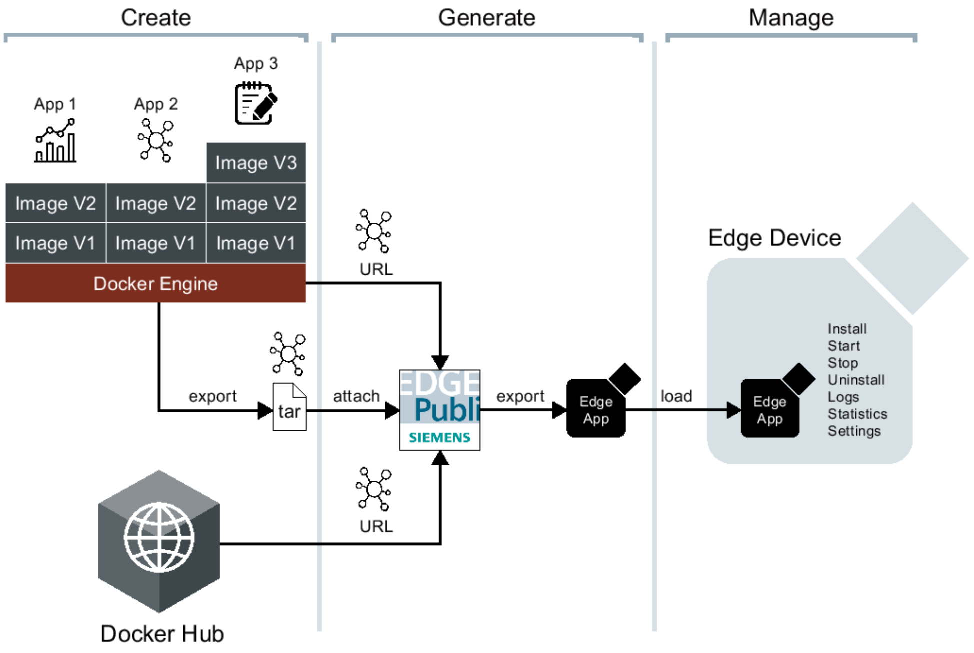
2 Engineering - Create and manage edge app

Perform the following steps to create an edge application for the Unified Comfort Panel:

General sequence

1. Create: Create Docker images
   → see chapter 2.2
2. Generate: Generate edge app with Industrial Edge Publisher
   → see chapter 2.3
3. Manage: Manage edge app on the Unified Comfort Panel
   → see chapter 2.4

Figure 2-1



Note Chapters 2.2.1 - 2.2.5 have already been carried out for the "SendMail App" in this application example.

If you want to implement this app, you can proceed directly with chapter 2.2.6.

Note You can find additional information on creating and managing edge apps in the Getting Started "Unified Comfort Panels for Industrial Edge".

https://support.industry.siemens.com/cs/ww/en/view/109773259

2.1 Installing Docker development environment

You need the Docker development environment in order to carry out the steps below.

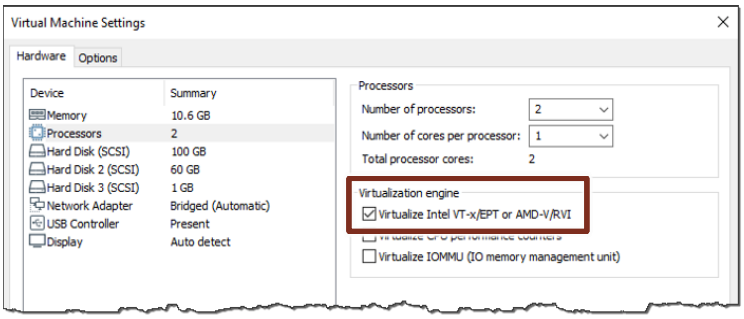
Installation and configuration of the service

1. Download the most recent version of "Desktop for Windows Installer" from this website:

   https://hub.docker.com/editions/community/docker-ce-desktop-windows/

   Note
   If you want to install Docker in a virtual environment (such as a virtual machine), you must enable "Forwarding virtualization technologies (Vt-x)" on the host PC in your VM's settings.

   Figure 2-2



Virtual Machine Settings

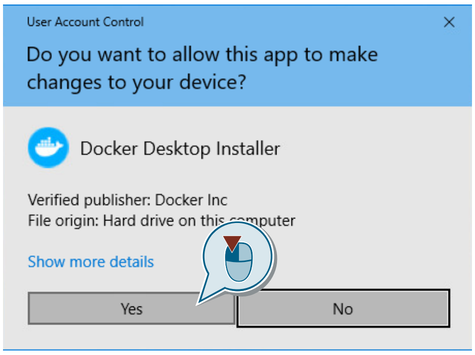
Device	Summary
Memory	10.6 GB
Processors	2
Hard Disk (SCSI)	100 GB
Hard Disk 2 (SCSI)	60 GB
Hard Disk 3 (SCSI)	1 GB
Network Adapter	Bridged (Automatic)
USB Controller	Present
Display	Auto detect

Processors
Number of processors: 2
Number of cores per processor: 1
Total processor cores: 2

Virtualization engine
☑ Virtualize Intel VT-x/EPT or AMD-V/RVI
☐ Virtualize CPU performance counters
☐ Virtualize IOMMU (IO memory management unit)

2. Once you have downloaded the application "Docker Desktop Installer.exe", launch it by double-clicking.

3. Confirm the message allowing the application to make changes to your device.



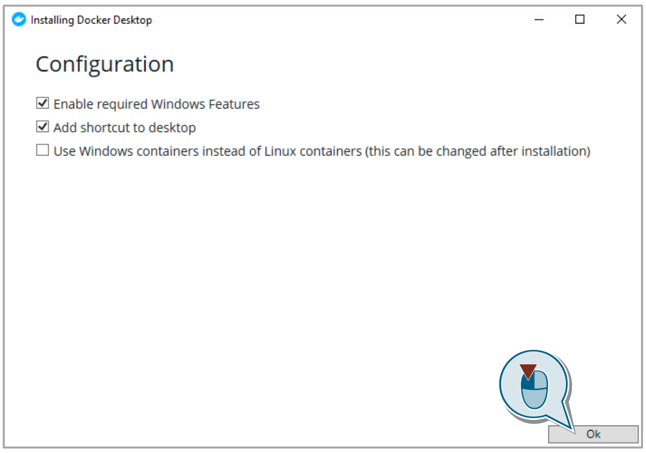
User Account Control

Do you want to allow this app to make changes to your device?

Docker Desktop Installer

Verified publisher: Docker Inc
File origin: Hard drive on this computer

4. In the "Configuration" step, click "OK".



Installing Docker Desktop

Configuration

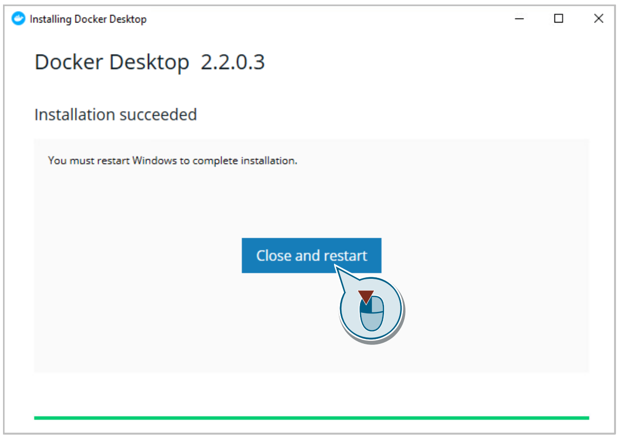
☑ Enable required Windows Features

☑ Add shortcut to desktop

□ Use Windows containers instead of Linux containers (this can be changed after installation)

Note The setting "Use Windows Containers instead of Linux containers (this can be changed after installation)" must be disabled, as the Unified Comfort Panel has a Linux operating system.

5. Confirm the installation and restart the computer.



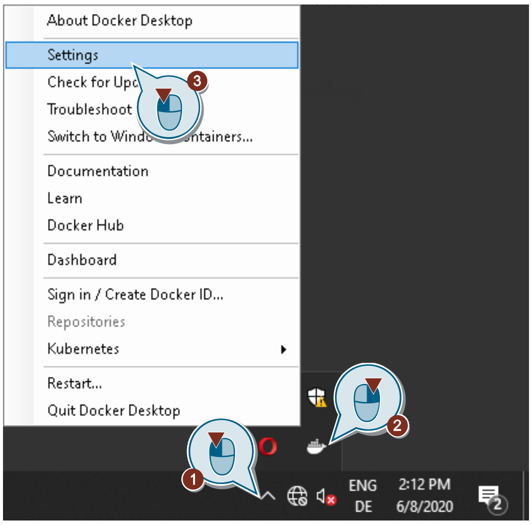
Installing Docker Desktop

Docker Desktop 2.2.0.3

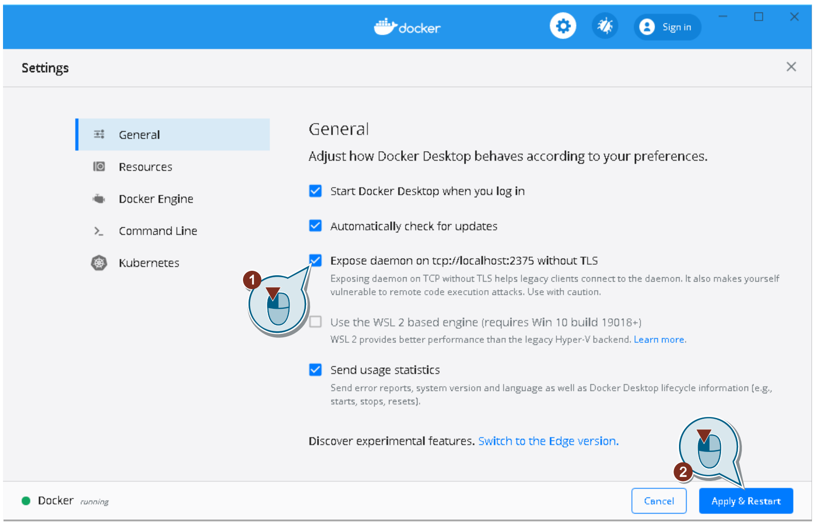
Installation succeeded

You must restart Windows to complete installation.

6. Open Docker settings
   - After restarting, wait until the Docker Service has started. Then open the advanced view of the Taskbar (1).
   - Right-click on the Docker icon (2).
   - In the context menu, click "Settings" (3).



7. Enable remote sharing
   - Enable the setting "Expose daemon on tcp://localhost:2375 without tls" for remote sharing of the Docker Engine (1).
   - Then click "Apply & Restart" to apply the settings and restart the Docker service. Restarting may take some time.



General

Adjust how Docker Desktop behaves according to your preferences.

☑ Start Docker Desktop when you log in

☑ Automatically check for updates

☑ Expose daemon on tcp://localhost:2375 without TLS
Exposing daemon on TCP without TLS helps legacy clients connect to the daemon. It also makes yourself vulnerable to remote code execution attacks. Use with caution.

□ Use the WSL 2 based engine (requires Win 10 build 19018+)
WSL 2 provides better performance than the legacy Hyper-V backend. Learn more.

☑ Send usage statistics
Send error reports, system version and language as well as Docker Desktop lifecycle information (e.g., starts, stops, resets).

Discover experimental features. Switch to the Edge version.

2.2 Create: Create Docker images

The application running inside the container can be created in any programming language. As an example, an application was created here that you can use to send emails with attachments over an SMTP server. The application is written in the JavaScript programming language.

2.2.1 Folder structure

For a better overview, first create a folder structure in Windows Explorer as shown in the figure below.

Figure 2-3

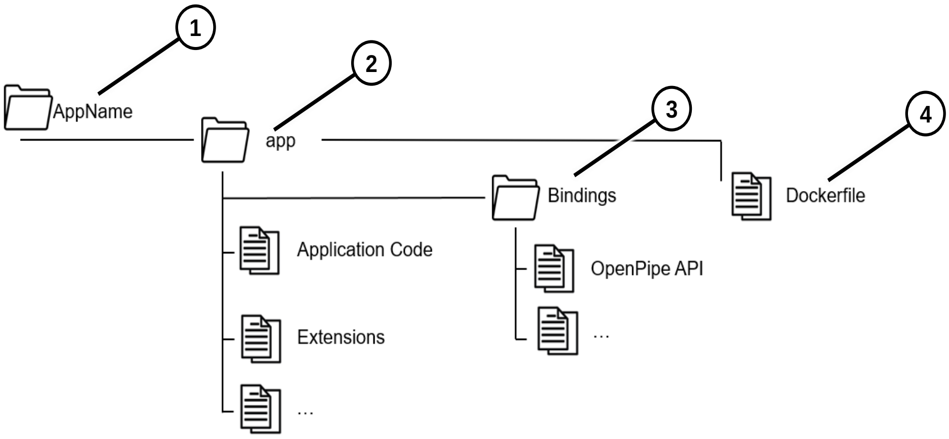


Table 2-1

No.	Folder	Description
1	"AppName"	Contains all relevant files for creating the app and the container
2	"app"	Contains the application proper and its dependencies, e.g. in the folder "Bindings" and the file "package.json"
3	"Bindings"	Contains the OpenPipe Comfort API for more easily establishing a connection to the Unified Comfort Panel runtime
4	"Dockerfile"	Contains the instruction list for creating the container

The user program can then be created. The section below describes in more detail the items specifically related to the functions of the Unified Comfort Panel.

Note

The "app" folder contains, among others, the "server.js" and "sendmail.js" files.

The "server.js" file starts a web server in which the console messages from "sendmail.js" are output in the browser. The state and error messages can be evaluated in this way.

To save resources you can also call "sendmail.js" directly in the "Dockerfile" (program without web console).

The "server.js" can also be used for other apps to enable console output without SSH rights. For this, the call of the ChildProcess (cp.spwan) must be adapted in the "server.js".

Note

Development environment, IDE

To develop the application, we recommend that you use an integrated development environment (IDE). In this example, we use the open source Visual Studio Code environment from Microsoft.

Visual Studio Code from Microsoft can be downloaded from the following link and installed:

https://code.visualstudio.com/

2.2.2 Configure connections OpenPipe connection

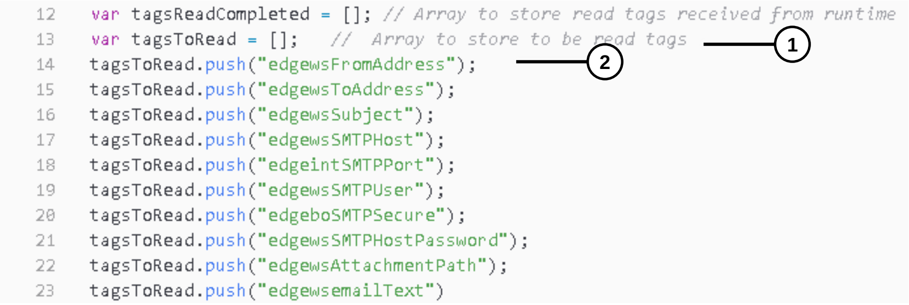
An OpenPipe connection is defined and utilized for data exchange between the edge application and the WinCC Unified runtime of the SIMATIC HMI Unified Comfort Panel.

Note

A ready-made API for the OpenPipe connection is available for ease of configuration. The necessary scripts can be found in the "Bindings" folder. This is supplied with WinCC Unified on DVD 2 in the folder "Openness".

The tags which will be exchanged with OpenPipe are defined first:

Figure 2-4



```
var tagsReadCompleted = []; // Array to store read tags received from runtime
var tagsToRead = [];   //  Array to store to be read tags
tagsToRead.push("edgewsFromAddress");
tagsToRead.push("edgewsToAddress");
tagsToRead.push("edgewsSubject");
tagsToRead.push("edgewsSMTPHost");
tagsToRead.push("edgeintSMTPPort");
tagsToRead.push("edgewsSMTPUser");
tagsToRead.push("edgeboSMTPSecure");
tagsToRead.push("edgewsSMTPHostPassword");
tagsToRead.push("edgewsAttachmentPath");
tagsToRead.push("edgewsemailText")
```

Note

Make sure that the tags are written exactly the same as in the TIA Portal project (case sensitive), otherwise no connection can be established.

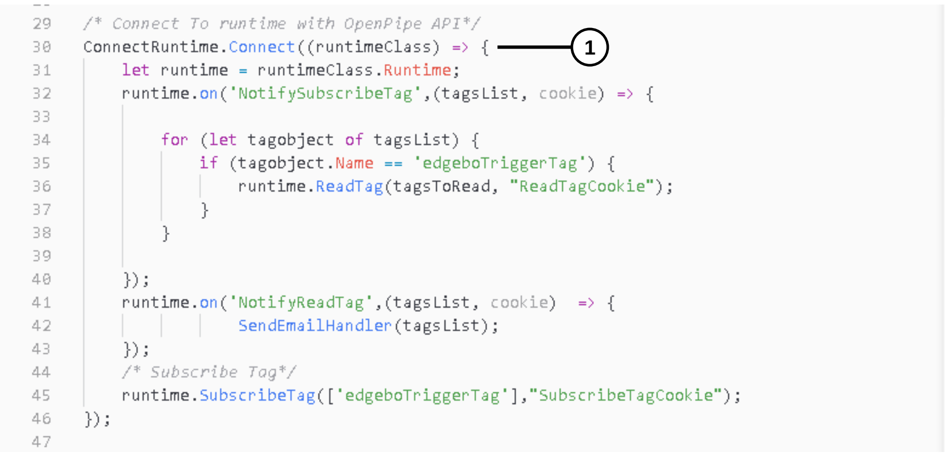
Table 2-2

No.	Description
①	Definition of the array for the tags to be read
②	Add the tag names to the array

Note

In the case of tag names that change, the list can be read externally using a script. To do this, the values are entered in a configuration file. This script is located in data storage outside of the container and is read again when the application is started.

Figure 2-5



```
/* Connect To runtime with OpenPipe API*/
ConnectRuntime.Connect((runtimeClass) => {
    let runtime = runtimeClass.Runtime;
    runtime.on('NotifySubscribeTag',(tagsList, cookie) => {

        for (let tagobject of tagsList) {
            if (tagobject.Name == 'edgeboTriggerTag') {
                runtime.ReadTag(tagsToRead, "ReadTagCookie");
            }
        }

    });
    runtime.on('NotifyReadTag',(tagsList, cookie)  => {
                SendEmailHandler(tagsList);
    });
    /* Subscribe Tag*/
    runtime.SubscribeTag(['edgeboTriggerTag'],"SubscribeTagCookie");
});
```

Table 2-3

No.	Description
1	Call of the ConnectRuntime class. This contains the OpenPipe API where the tags from the list "tagsList" are read. The trigger tag is read in the event of change in the Unified Comfort Panel runtime (subscribe).

This instruction establishes a connection to the runtime by means of OpenPipe, via which the tags are exchanged. The names of the tags to be read are in the "tagsList" list. The OpenPipe system functions such as "ReadTag" and "SubscribeTag" are part of the Unified Comfort Panel API.

2.2.3 Programming the email function

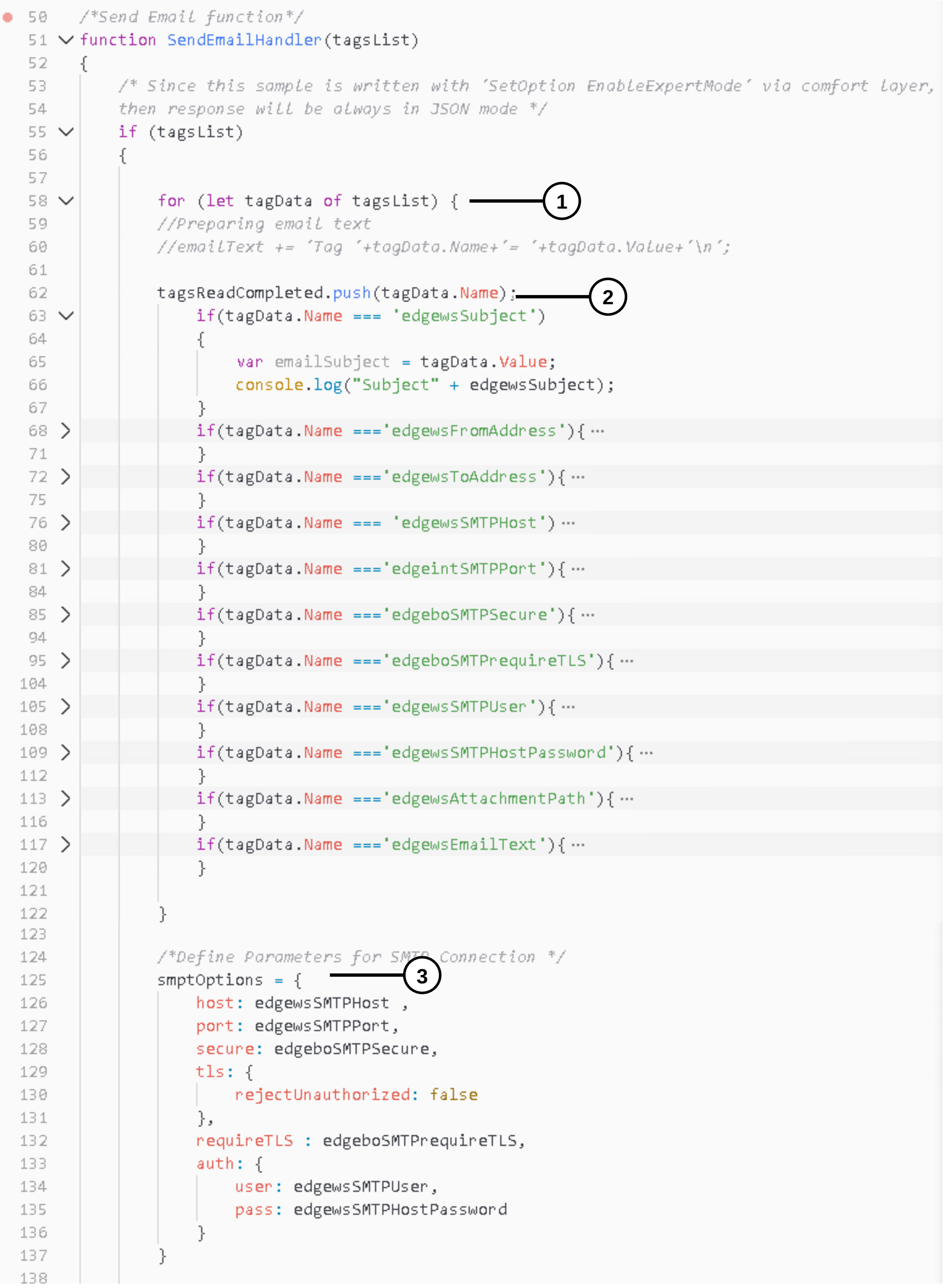
Once the tag handoff is defined, the email function is implemented. For nodeJS there are multiple modules which offer an SMTP function. This example uses the module "nodemailer".

Note

Please refer to the following link for more information about nodemailer:

https://nodemailer.com/about/

Figure 2-6

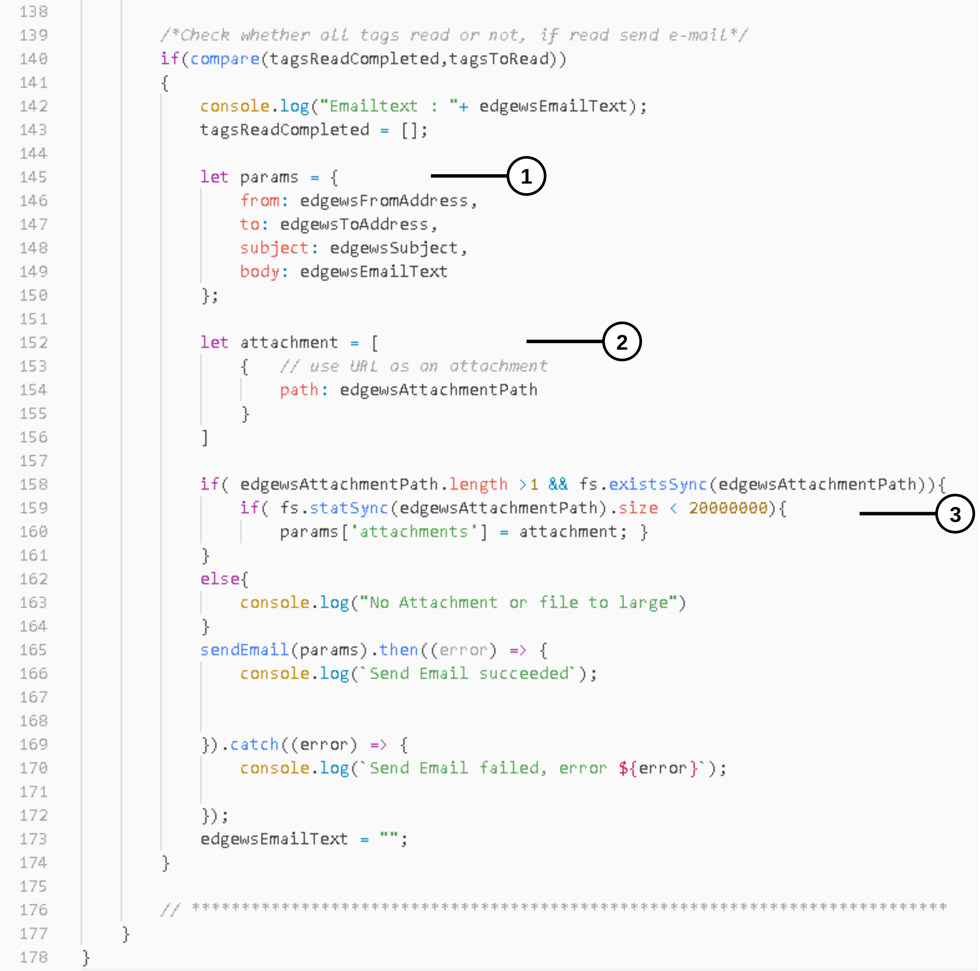


```
/*Send Email function*/
function SendEmailHandler(tagsList)
{
    /* Since this sample is written with 'SetOption EnableExpertMode' via comfort layer,
    then response will be always in JSON mode */
    if (tagsList)
    {

        for (let tagData of tagsList) {
        //Preparing email text
        //emailText += 'Tag '+tagData.Name+'= '+tagData.Value+'\n';

        tagsReadCompleted.push(tagData.Name);
            if(tagData.Name === 'edgewsSubject')
            {
                var emailSubject = tagData.Value;
                console.log("Subject" + edgewsSubject);
            }
            if(tagData.Name ==='edgewsFromAddress'){ ···
            }
            if(tagData.Name ==='edgewsToAddress'){ ···
            }
            if(tagData.Name === 'edgewsSMTPHost') ···
            }
            if(tagData.Name ==='edgeintSMTPPort'){ ···
            }
            if(tagData.Name ==='edgeboSMTPSecure'){ ···
            }
            if(tagData.Name ==='edgeboSMTPrequireTLS'){ ···
            }
            if(tagData.Name ==='edgewsSMTPUser'){ ···
            }
            if(tagData.Name ==='edgewsSMTPHostPassword'){ ···
            }
            if(tagData.Name ==='edgewsAttachmentPath'){ ···
            }
            if(tagData.Name ==='edgewsEmailText'){ ···
            }

        }

        /*Define Parameters for SMTP Connection */
        smptOptions = {
            host: edgewsSMTPHost ,
            port: edgewsSMTPPort,
            secure: edgeboSMTPSecure,
            tls: {
                rejectUnauthorized: false
            },
            requireTLS : edgeboSMTPrequireTLS,
            auth: {
                user: edgewsSMTPUser,
                pass: edgewsSMTPHostPassword
            }
        }

```

Table 2-4

No.	Description
1	Counter loop for the individual tag names to identify the tag in the JSON object.
2	Value assignment for local tags.
3	Assignment of the connection parameters for the Nodemailer module in order to create a connection to the SMTP (outbound mail server). (A user and password are usually required for this.) The parameters for port and security settings are defined as well.

Figure 2-7

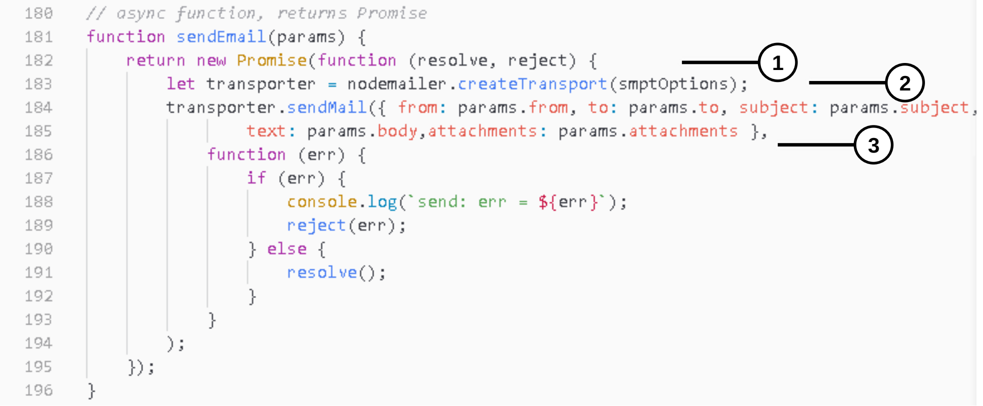


```

        /*Check whether all tags read or not, if read send e-mail*/
        if(compare(tagsReadCompleted,tagsToRead))
        {
            console.log("Emailtext : "+ edgewsEmailText);
            tagsReadCompleted = [];

            let params = {
                from: edgewsFromAddress,
                to: edgewsToAddress,
                subject: edgewsSubject,
                body: edgewsEmailText
            };

            let attachment = [
                {   // use URL as an attachment
                    path: edgewsAttachmentPath
                }
            ]

            if( edgewsAttachmentPath.length >1 && fs.existsSync(edgewsAttachmentPath)){
                if( fs.statSync(edgewsAttachmentPath).size < 20000000){
                    params['attachments'] = attachment; }
            }
            else{
                console.log("No Attachment or file to large")
            }
            sendEmail(params).then((error) => {
                console.log(`Send Email succeeded`);


            }).catch((error) => {
                console.log(`Send Email failed, error ${error}`);

            });
            edgewsEmailText = "";
        }

        // ******************************************************************************
    }
}
```

Table 2-5

No.	Description
1	Assignment for the message parameters
2	Definition of the attachment (path of the file which will be sent)
3	Attachment validation (whether an attachment is present; and the size of the attachment)

Figure 2-8



```
// async function, returns Promise
function sendEmail(params) {
    return new Promise(function (resolve, reject) {
        let transporter = nodemailer.createTransport(smptOptions);
        transporter.sendMail({ from: params.from, to: params.to, subject: params.subject,
                text: params.body,attachments: params.attachments },
            function (err) {
                if (err) {
                    console.log(`send: err = ${err}`);
                    reject(err);
                } else {
                    resolve();
                }
            }
        );
    });
}
```

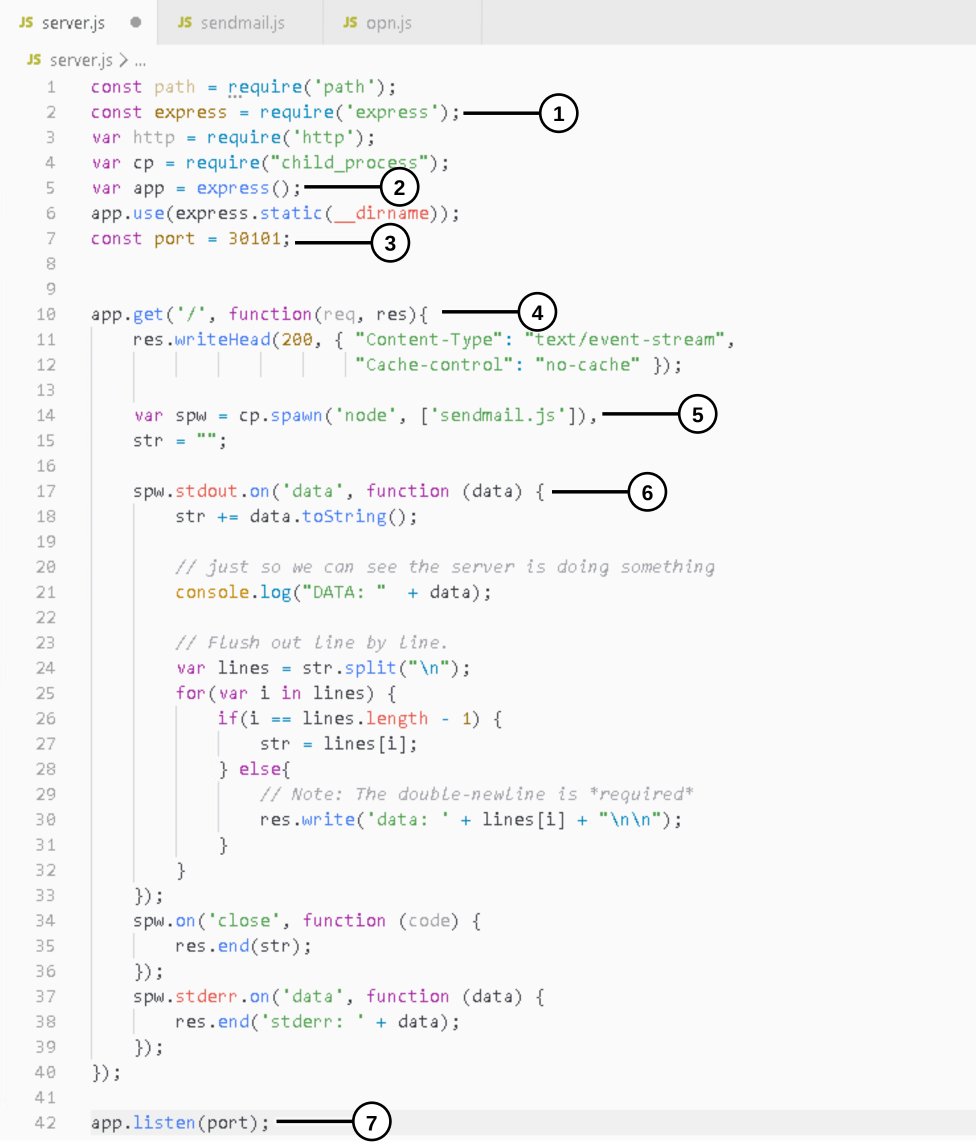
Table 2-6

No.	Description
1	Create the Promise for sending email (the Promise object constitutes a representation of a prospective execution (or failure) of an asynchronous operation and the ensuing results).
2	Create the object for sending
3	Handoff of the parameters

2.2.4 Outputting container logs via web interface

A redirect to a website is implemented in order to receive an output of the container log. Here, the web server is also running in the container. A simple way of creating a website is to use a framework such as Express, which has already defined basic tasks like routing.

Figure 2-9



```
1   const path = require('path');
2   const express = require('express');
3   var http = require('http');
4   var cp = require("child_process");
5   var app = express();
6   app.use(express.static(__dirname));
7   const port = 30101;
8
9
10  app.get('/', function(req, res){
11      res.writeHead(200, { "Content-Type": "text/event-stream",
12                           "Cache-control": "no-cache" });
13
14      var spw = cp.spawn('node', ['sendmail.js']),
15      str = "";
16
17      spw.stdout.on('data', function (data) {
18          str += data.toString();
19
20          // just so we can see the server is doing something
21          console.log("DATA: "  + data);
22
23          // Flush out line by line.
24          var lines = str.split("\n");
25          for(var i in lines) {
26              if(i == lines.length - 1) {
27                  str = lines[i];
28              } else{
29                  // Note: The double-newline is *required*
30                  res.write('data: ' + lines[i] + "\n\n");
31              }
32          }
33      });
34      spw.on('close', function (code) {
35          res.end(str);
36      });
37      spw.stderr.on('data', function (data) {
38          res.end('stderr: ' + data);
39      });
40  });
41
42  app.listen(port);
```

Table 2-7

No.	Description
1	Call of the Express web frontend -> for documentation, refer to: https://expressjs.com/en/guide/routing.html.
2	Declaration of Express.
3	Definition of the port where the web server is reachable (starting at port 30000 in order to prevent errors with other applications).
4	Definition of the route http://localhost/ → writes the output from stdout to a header and sends it to the client.
5	Starts a child process with the SendMail application. The associated declaration happens in line 4.

No.	Description
6	Redirects the "stdout" to the Express framework.
7	Starts the web server.

2.2.5 Changing application into container format

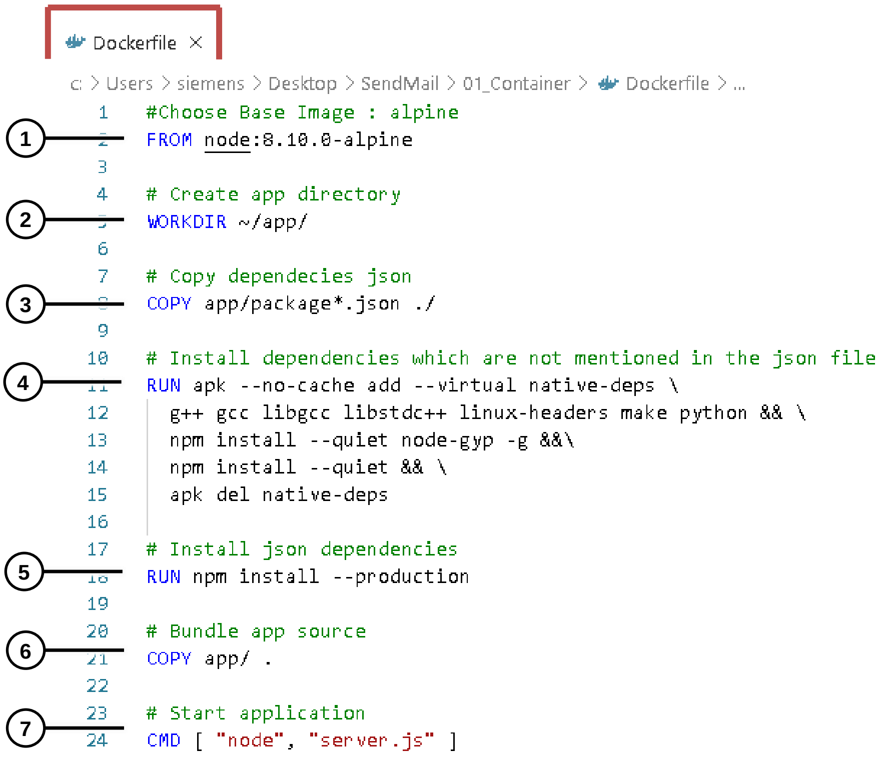
Create container

Once the application has been written, it is packaged into a container which contains all information, programs and dependencies. This makes it possible to easily port this container image.

Instruction list for creating a container

All commands for creating a Docker image are in the instruction list, also known as the Docker file. This image contains all dependencies in order to run the application in the container. These can be, for instance, a Python Interpreter, or modules that are necessary for starting the application. There are often ready-made images containing numerous software packages which obviate additional installations. However, these images often require a lot of resources from the Unified Comfort Panel. In this case it is usually better to take a lightweight base image (such as Alpine, Ubuntu Base) in order to achieve a small, resource-saving image. The image size is a major factor in the final app size.

Figure 2-10


Dockerfile ×
c: > Users > siemens > Desktop > SendMail > 01_Container > Dockerfile > ...
1
2
3
4
5
6
7

```
#Choose Base Image : alpine
FROM node:8.10.0-alpine

# Create app directory
WORKDIR ~/app/

# Copy dependecies json
COPY app/package*.json ./

# Install dependencies which are not mentioned in the json file
RUN apk --no-cache add --virtual native-deps \
  g++ gcc libgcc libstdc++ linux-headers make python && \
  npm install --quiet node-gyp -g &&\
  npm install --quiet && \
  apk del native-deps

# Install json dependencies
RUN npm install --production

# Bundle app source
COPY app/ .

# Start application
CMD [ "node", "server.js" ]
```

Table 2-8

No.	Command	Explanation
①	FROM	A nodeJS image based on the Alpine distribution is used as a base image. It is very resource-light because it only needs a few megabytes of memory.
②	WORKDIR	Defines the directory where the commands are executed.
③	COPY	The package.json and package-lock.json files from the local directory /app are copied to the working directory.
④	RUN	Installs other applications such as gcc and python.
⑤	RUN	Installs the dependencies for package.json.
⑥	COPY	Copies the application to the working directory.
⑦	CMD	Starts the application.

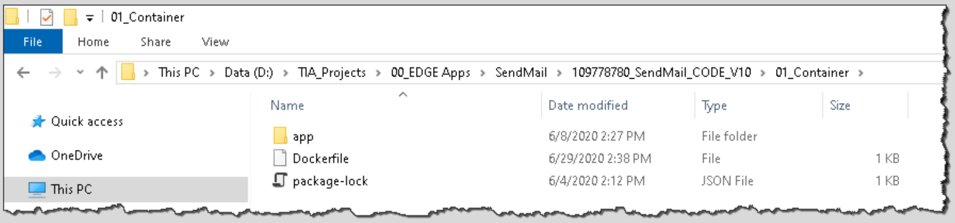
2.2.6 Creating the image and testing the container

Using the Docker command "build", the image is then generated according to the specifications of the Docker file.

Note

If you wish to use the SendMail app from this application example, then first download the "SendMail App" file and extract it.

Once you have extracted the file, you will find all former files in the "01_Container" folder, including the file "Dockerfile".


01_Container
File
Home
Share
View
This PC › Data (D:) › TIA_Projects › 00_EDGE Apps › SendMail › 109778780_SendMail_CODE_V10 › 01_Container ›
Quick access
OneDrive
This PC

Name	Date modified	Type	Size
app	6/8/2020 2:27 PM	File folder	
Dockerfile	6/29/2020 2:38 PM	File	1 KB
package-lock	6/4/2020 2:12 PM	JSON File	1 KB

Proceed as follows to do this:

1. Start a command line, for instance with CMD.
2. In the command line, switch to the folder with the file "Dockerfile".
3. Then execute the command

```
docker build -t sendmail .
```

Docker "build" starts the build process, the parameter "-t" specifies the freely selectable name of the image, which in turn specifies the directory where the Docker file is located (here, in the same directory).

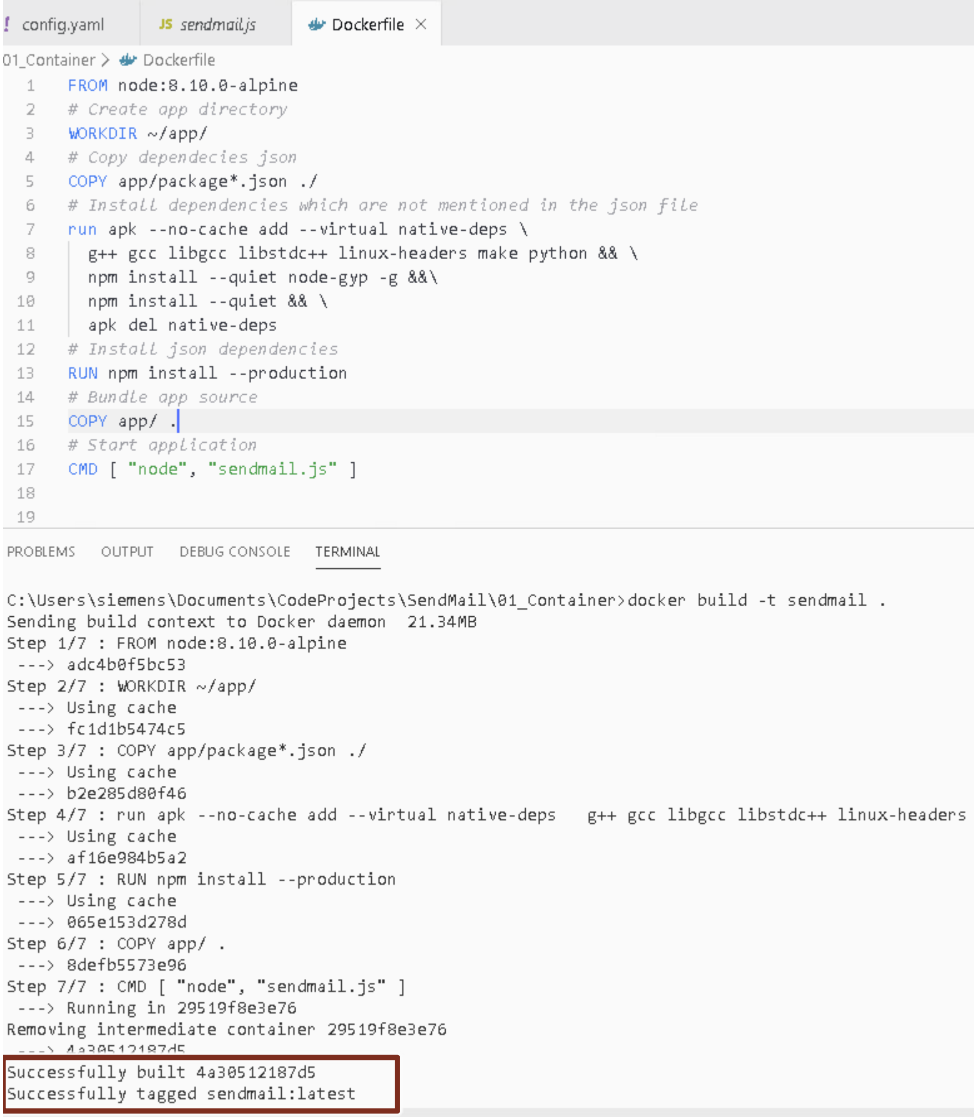
Note

An internet connection is required for the build process. The files needed for the installation are downloaded automatically.

Result

The result in Visual Studio Code is shown below.

Figure 2-11



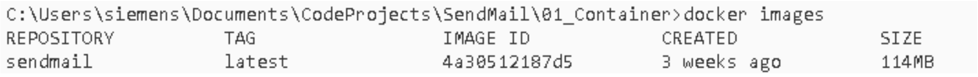
```
FROM node:8.10.0-alpine
# Create app directory
WORKDIR ~/app/
# Copy dependecies json
COPY app/package*.json ./
# Install dependences which are not mentioned in the json file
run apk --no-cache add --virtual native-deps \
  g++ gcc libgcc libstdc++ linux-headers make python && \
  npm install --quiet node-gyp -g &&\
  npm install --quiet && \
  apk del native-deps
# Install json dependencies
RUN npm install --production
# Bundle app source
COPY app/ .
# Start application
CMD [ "node", "sendmail.js" ]
```

```
C:\Users\siemens\Documents\CodeProjects\SendMail\01_Container>docker build -t sendmail .
Sending build context to Docker daemon  21.34MB
Step 1/7 : FROM node:8.10.0-alpine
 ---> adc4b0f5bc53
Step 2/7 : WORKDIR ~/app/
 ---> Using cache
 ---> fc1d1b5474c5
Step 3/7 : COPY app/package*.json ./
 ---> Using cache
 ---> b2e285d80f46
Step 4/7 : run apk --no-cache add --virtual native-deps   g++ gcc libgcc libstdc++ linux-headers
 ---> Using cache
 ---> af16e984b5a2
Step 5/7 : RUN npm install --production
 ---> Using cache
 ---> 065e153d278d
Step 6/7 : COPY app/ .
 ---> 8defb5573e96
Step 7/7 : CMD [ "node", "sendmail.js" ]
 ---> Running in 29519f8e3e76
Removing intermediate container 29519f8e3e76
 ---> 4a30512187d5
Successfully built 4a30512187d5
Successfully tagged sendmail:latest
```

Once the build process has completed successfully, the result with the image number appears in the output.

Using the "docker images" command, the existing images can be output:

Figure 2-12



```
C:\Users\siemens\Documents\CodeProjects\SendMail\01_Container>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
sendmail            latest              4a30512187d5        3 weeks ago         114MB
```

Note

In order to simplify the development and test environment, the application and the container can be tested in conjunction with a WinCC Unified PC runtime. The OpenPipe function of the Unified PC runtime behaves like an OpenPipe function on the SIMATIC HMI Unified Comfort Panel. However, attention must be paid to correct addressing.

Additional information

The following table lists common Docker commands and their meaning.

Table 2-9

No.	Command	Description
1.	Docker build	Creates an image from a Docker file
2.	Docker images	Lists existing images
3.	Docker run	Starts a container
4.	Docker ps	Lists running containers
5.	Docker exec	Executes a command in a running container
6.	Docker rm	Deletes a container
7.	Docker rmi	Deletes an existing image
8.	Docker stop	Stops a running container

Note Further information about the commands can be found at the following link:

https://docs.docker.com/engine/reference/commandline/docker/

2.3 Generate: Generate edge app with Industrial Edge Publisher

2.3.1 Install Siemens Industrial Edge Publisher

You can download and install the Siemens Industrial Edge Publisher for free from the download "Siemens Industrial Edge Publisher" in the Siemens Industry Online Portal.

https://support.industry.siemens.com/cs/ww/en/view/109778875

Note For more information on Industrial Edge Publisher, refer to the associated programming manual:

https://support.industry.siemens.com/cs/ww/en/view/109778824

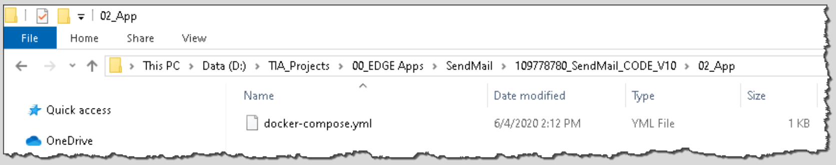
Overview and settings of the Publisher

Industrial Edge Publisher helps create a complete service (the edge app) from the container image and other information. The service contains all parameters pertaining to network connection, memory size and resource allocation.

You can use the Industrial Edge Publisher in two ways:

1. via the Wizard and the created container images, or
2. via a finished YAML file for Docker-Compose applications.

Note The finished YAML file for Docker-Compose applications can be found in the application example in the file "SendeMail App". Once you have downloaded and extracted the file, you will see the file in the folder "02_App".



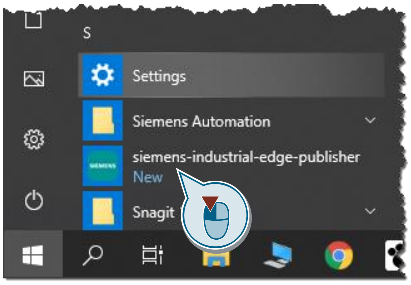
This PC › Data (D:) › TIA_Projects › 00_EDGE Apps › SendMail › 109778780_SendMail_CODE_V10 › 02_App

Name	Date modified	Type	Size
docker-compose.yml	6/4/2020 2:12 PM	YML File	1 KB

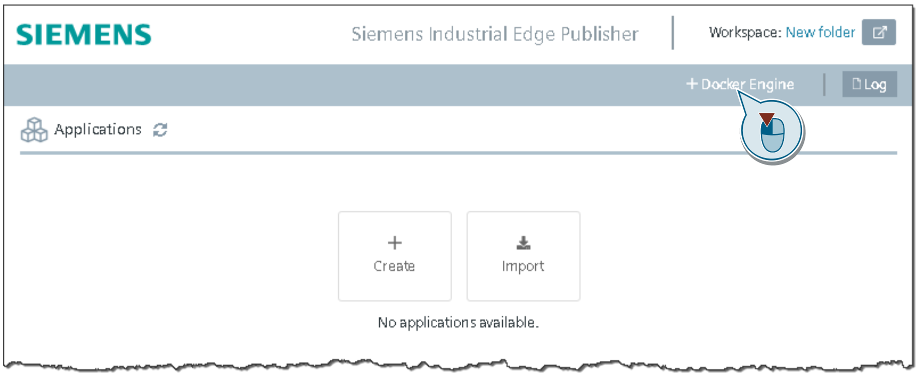
2.3.2 Generating the SendMail app with Siemens Industrial Edge Publisher

This chapter will demonstrate how to generate an edge app from the container images.

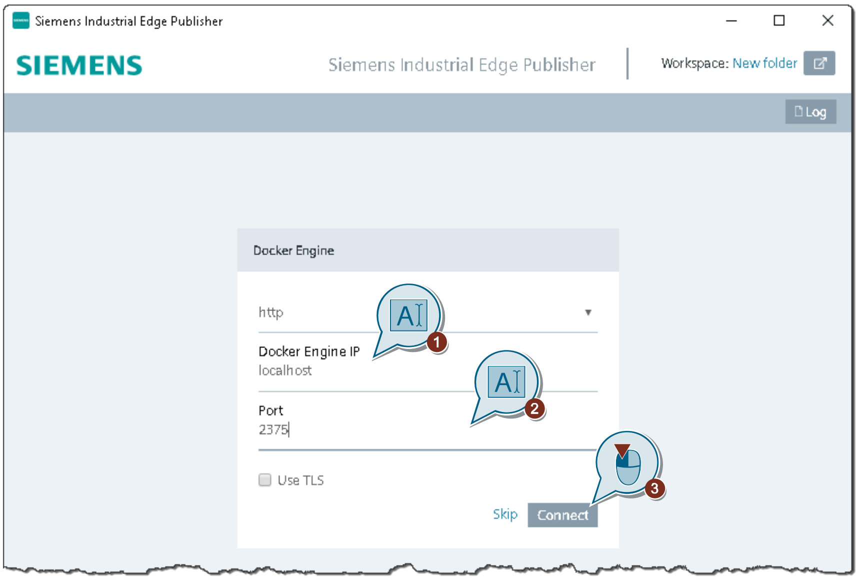
1. Start the Siemens Industrial Edge Publisher from the Windows Start Menu.



2. First click "+ Docker Engine" to define a Docker Engine and get access to the container images.



3. Connect Docker Engine
   - Enter the IP address or DNS name. If the Docker Engine is on the same computer, you can enter "localhost" (1).
   - For port, enter 2375 (default port) (2).
   - Then click "Connect" (3).


Siemens Industrial Edge Publisher
SIEMENS
Siemens Industrial Edge Publisher
Workspace: New folder
Log
Docker Engine
1
2
3
Skip
Connect

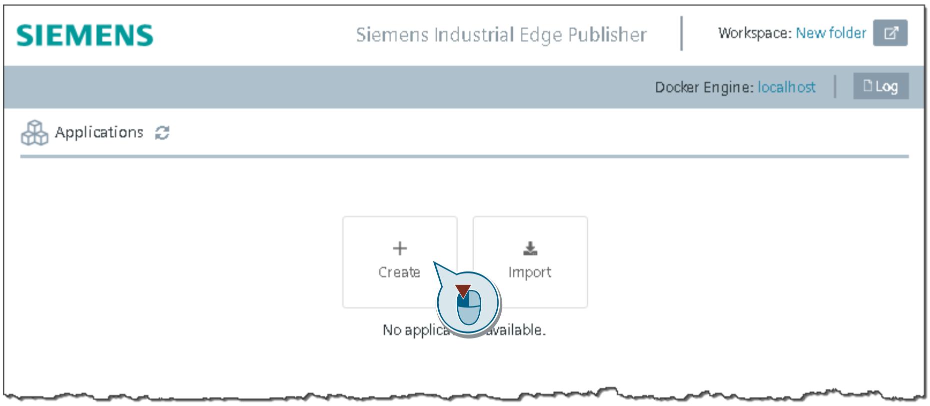
http

Docker Engine IP
localhost

Port
2375

□ Use TLS

4. Then click "Create" to create a new edge app.

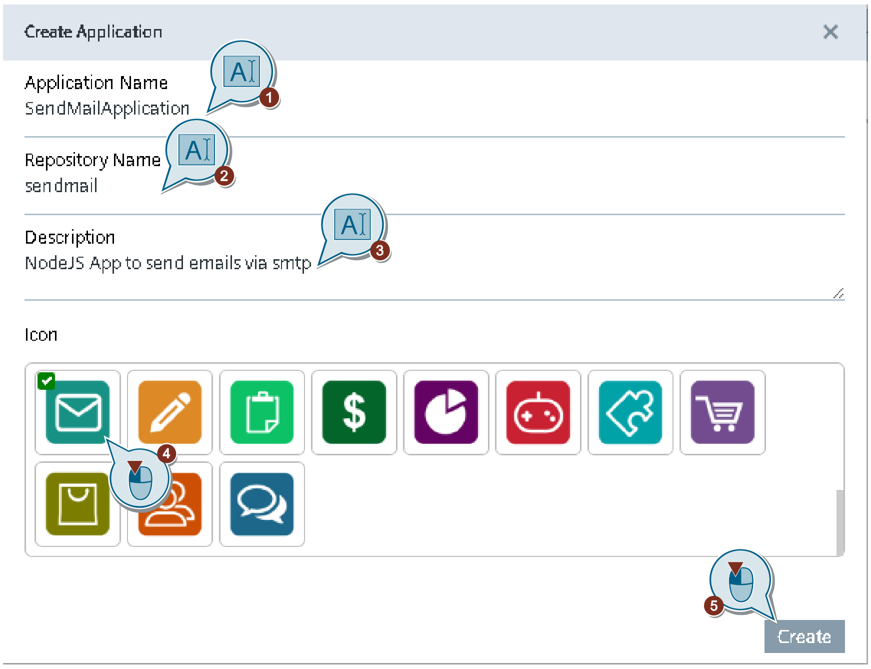


SIEMENS Siemens Industrial Edge Publisher | Workspace: New folder

Docker Engine: localhost | Log

Applications

5. Edit app properties
   - Enter the name ("Application Name") (1), "Repository Name" (2) and the description ("Description") (3) for the app.
   - Then select an "Icon" for the app (4).
   - Then click Create (5).



Create Application

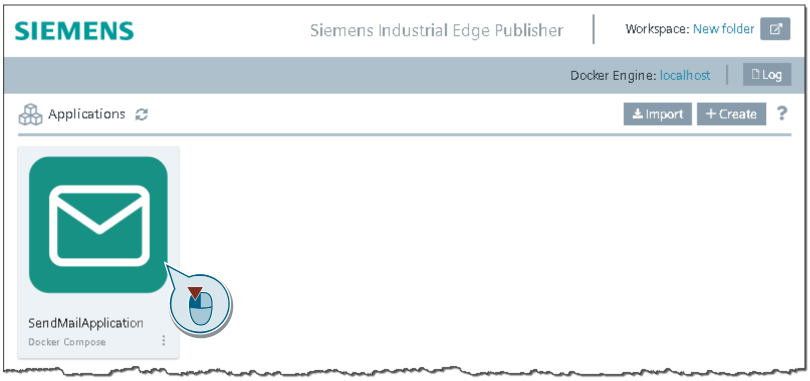
Application Name
SendMailApplication

Repository Name
sendmail

Description
NodeJS App to send emails via smtp

Icon

6. Open the app you created.

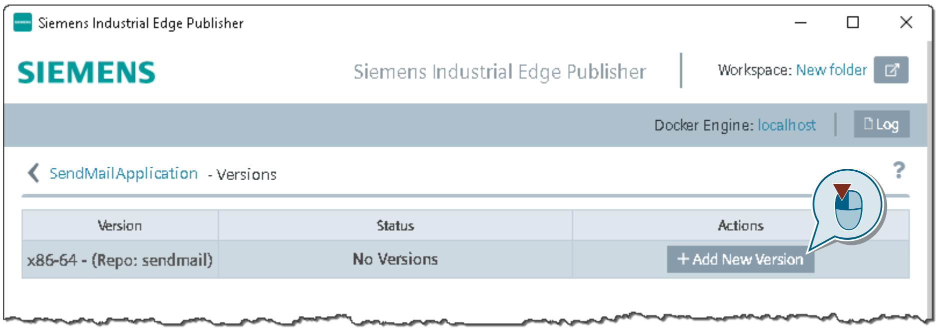


SIEMENS Siemens Industrial Edge Publisher | Workspace: New folder

Docker Engine: localhost | Log

Applications

7. Click "+ Add New Version" to add a new version to the app.



Siemens Industrial Edge Publisher

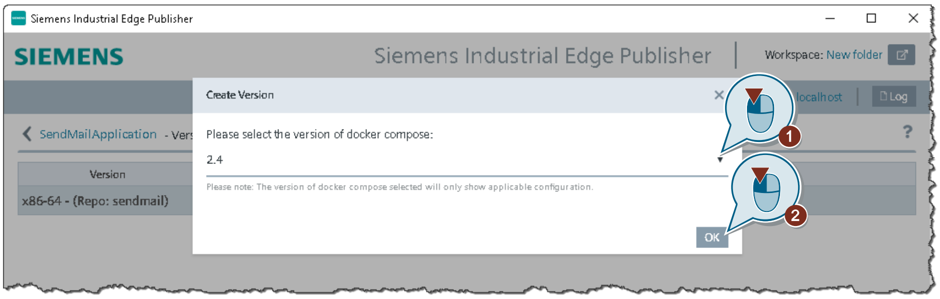
SIEMENS Siemens Industrial Edge Publisher Workspace: New folder

Docker Engine: localhost Log

SendMailApplication - Versions

Version	Status	Actions
x86-64 - (Repo: sendmail)	No Versions	+ Add New Version

8. Then select a version and click "OK".



Siemens Industrial Edge Publisher

SIEMENS Siemens Industrial Edge Publisher Workspace: New folder

Create Version

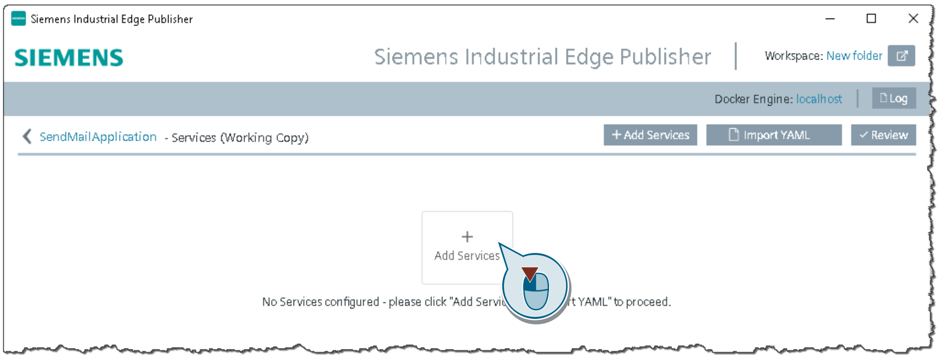
Please select the version of docker compose:

2.4

Please note: The version of docker compose selected will only show applicable configuration.

OK

9. In the next step, click "Add Service".



Siemens Industrial Edge Publisher

SIEMENS Siemens Industrial Edge Publisher Workspace: New folder

Docker Engine: localhost Log

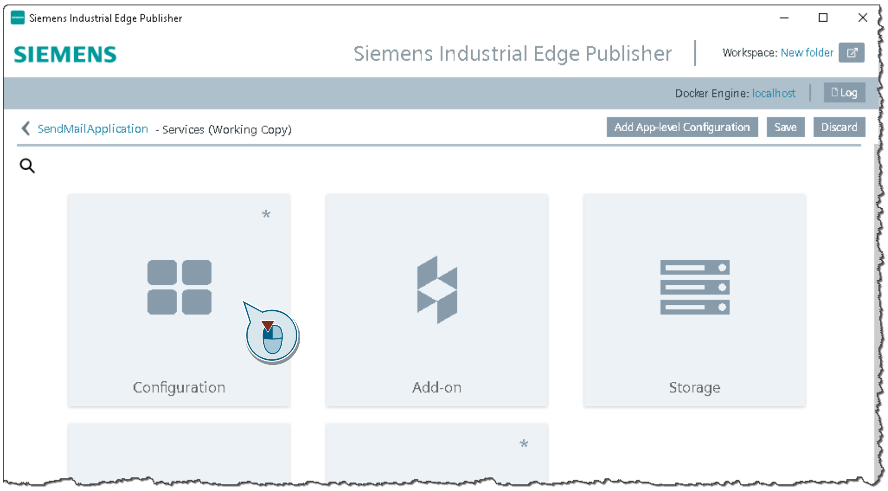
SendMailApplication - Services (Working Copy) + Add Services Import YAML ✓ Review

+ Add Services

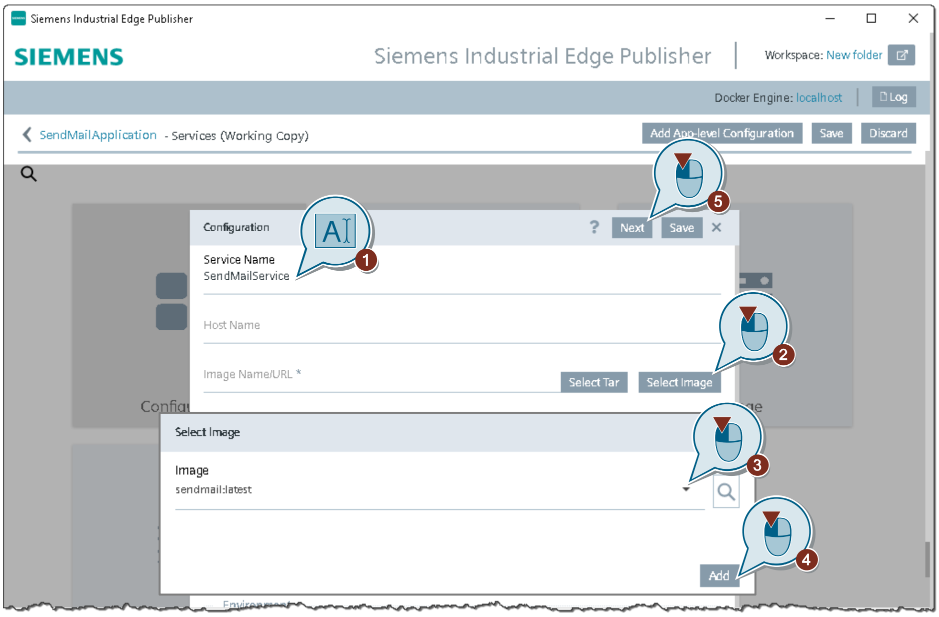
No Services configured - please click "Add Servi[...]t YAML" to proceed.

Optionally, in this step you can import the finished YAML file for Docker-Compose applications.

10. In the next step, start the configuration wizard.


Siemens Industrial Edge Publisher
SIEMENS
Siemens Industrial Edge Publisher
Workspace: New folder
Docker Engine: localhost
Log
SendMailApplication - Services (Working Copy)
Add App-level Configuration
Save
Discard
Configuration
Add-on
Storage

11. Select Docker image
    - First enter a name for the service (1).
    - Then, under "Image Name", click "Select image" (2).
    - Select the image from the dropdown list (3) and click "Add" to add it (4).
    - Then click "Next" (5).

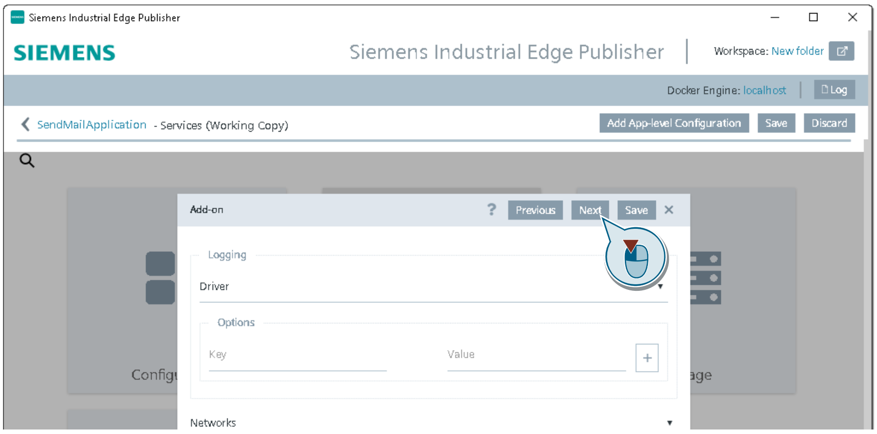


Service Name: SendMailService

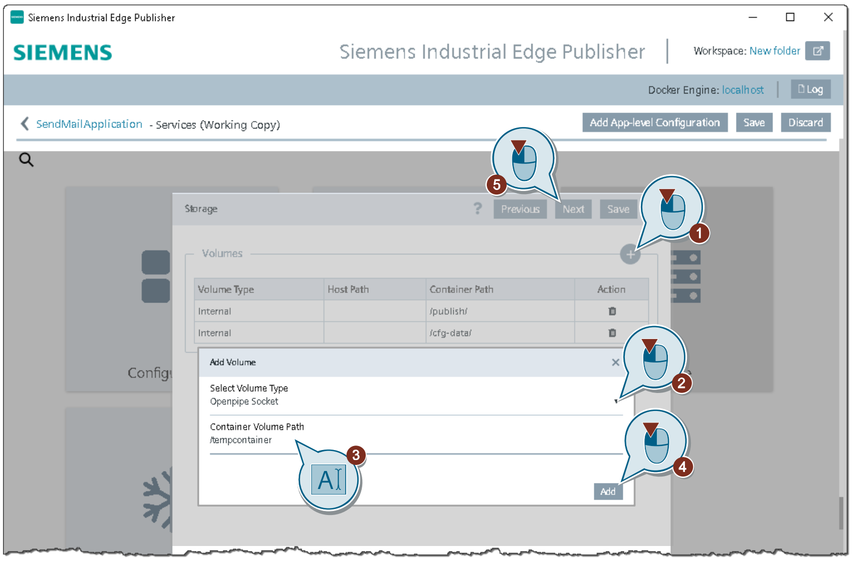
Image: sendmail:latest

Note The "Select Image" button is only visible if a Docker Engine has been connected.

12. In the "Add-on" step, click "Next", as no configuration steps are necessary here.


Siemens Industrial Edge Publisher
SIEMENS
Siemens Industrial Edge Publisher
Workspace: New folder
Docker Engine: localhost
Log
SendMailApplication - Services (Working Copy)
Add App-level Configuration
Save
Discard
Add-on
Previous
Next
Save
Logging
Driver
Options
Key
Value
Networks
Config
age

13. Create an OpenPipe connection
    - Click the "+" button (1).
    - Then select "Openpipe Socket" from the dropdown list (2).
    - Enter the container path of the connection (stored in the bindings). In this example it is `/tempcontainer` (3).
    - Then click "Add" (4).


Siemens Industrial Edge Publisher
SIEMENS
Siemens Industrial Edge Publisher
Workspace: New folder
Docker Engine: localhost
Log
SendMailApplication - Services (Working Copy)
Add App-level Configuration
Save
Discard
Storage
Previous
Next
Save
Volumes
Config
1
2
3
4
5

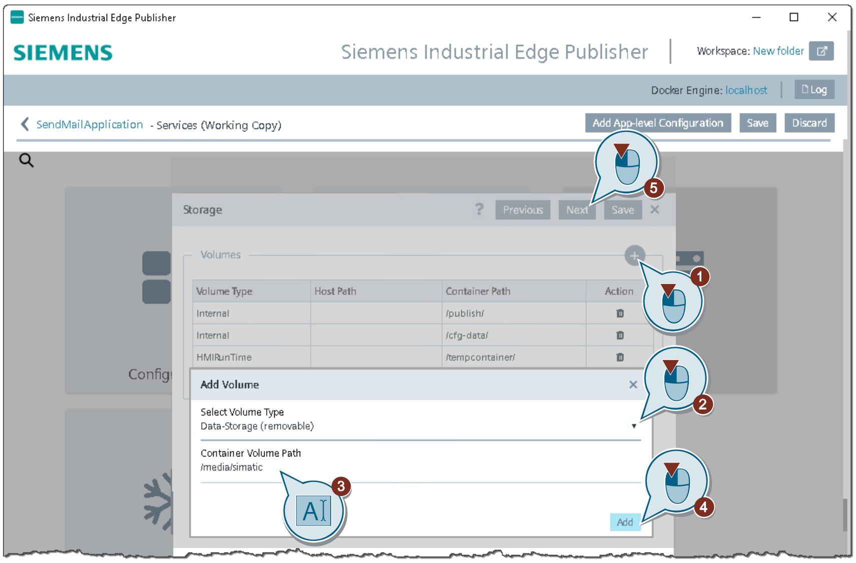
Volume Type	Host Path	Container Path	Action
Internal		/publish/	
Internal		/cfg-data/	

Add Volume

Select Volume Type: Openpipe Socket

Container Volume Path: /tempcontainer

14. Integrating USB flash drive
    - Click the "+" button again (1).
    - Then select "Data Storage (removable)" from the dropdown list (2).
    - Enter the path of the USB drive, `/media/simatic` (3).
    - Then click "Add" (4).
    - Click on "Next" (5).


Siemens Industrial Edge Publisher
SIEMENS
Siemens Industrial Edge Publisher
Workspace: New folder
Docker Engine: localhost
Log
SendMailApplication - Services (Working Copy)
Add App-level Configuration
Save
Discard
Storage
Previous
Next
Save
Volumes
Config
1
2
3
4
5

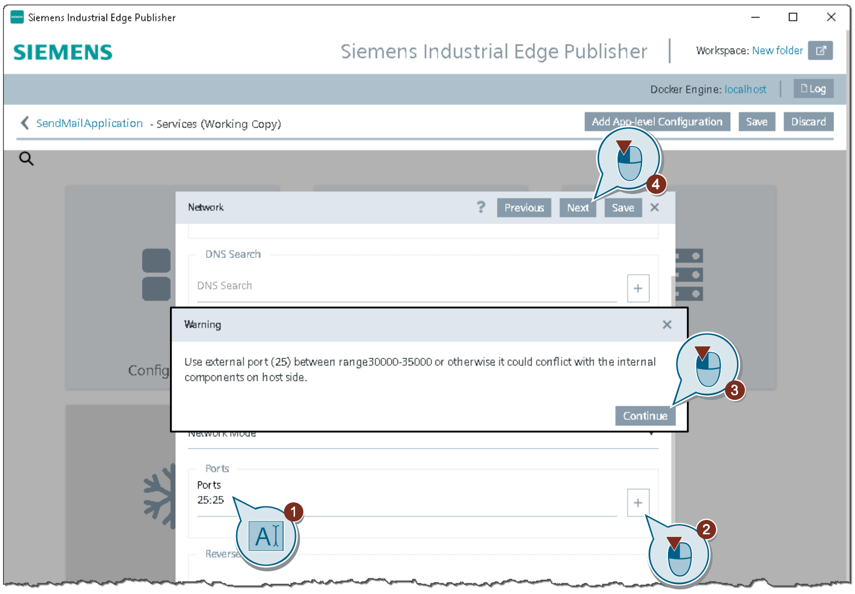
Volume Type	Host Path	Container Path	Action
Internal		/publish/	
Internal		/cfg-data/	
HMIRunTime		/tempcontainer/	

Add Volume

Select Volume Type: Data-Storage (removable)

Container Volume Path: /media/simatic

15. Define ports for network setting, for app communication
    - Enter the port number "25:25" (1).
    - Then click "+" (2).
    - Confirm the warning by clicking "Continue" (3).
    - Repeat the steps for the ports "465:465", "587:587" and "30101:30101".
    - Then click "next" (4).



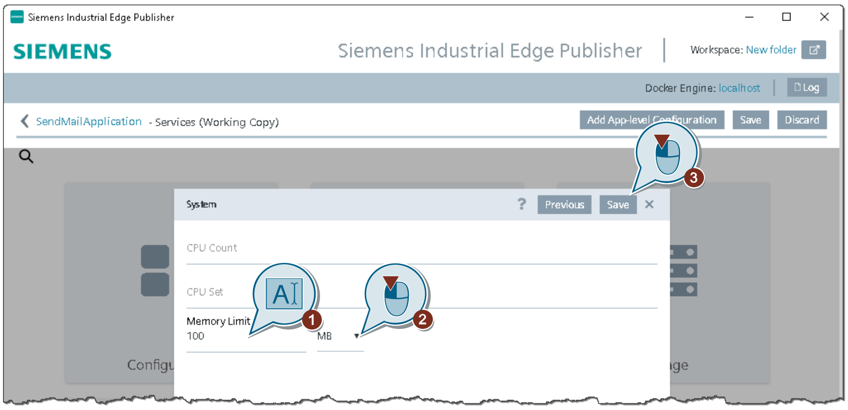
Use external port (25) between range30000-35000 or otherwise it could conflict with the internal components on host side.

Ports: 25:25

Note

Ports 25, 465, 587 are the default ports for SMTP communication. They are strictly defined ports and therefore they must also be used in the application. Switching to a port greater than 30000 is thus not possible.
Port 30101 is configured for use of the web server (see also chapter 2.2.4). The ports in use have no impact on the runtime.

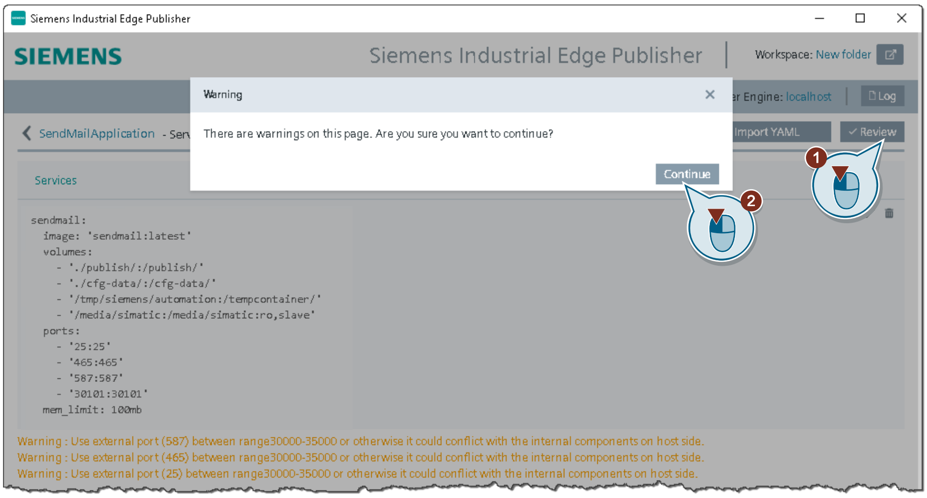
16. Define system limit of the app
    - For the memory limit, enter 100 (1) and then change the unit to "MB" (2).
    - Then click on "Save" (3).


Siemens Industrial Edge Publisher
SIEMENS
Siemens Industrial Edge Publisher
Workspace: New folder
Docker Engine: localhost
Log
SendMailApplication - Services (Working Copy)
Add App-level Configuration
Save
Discard
System
Previous
Save
CPU Count
CPU Set
Configu
ge
1
2
3

Memory Limit: 100 MB

Once the configuration wizard is complete, you will receive an overview of the settings in YAML format.

17. In the next step, click "Review" (1) and confirm the warning with "Continue" (2).



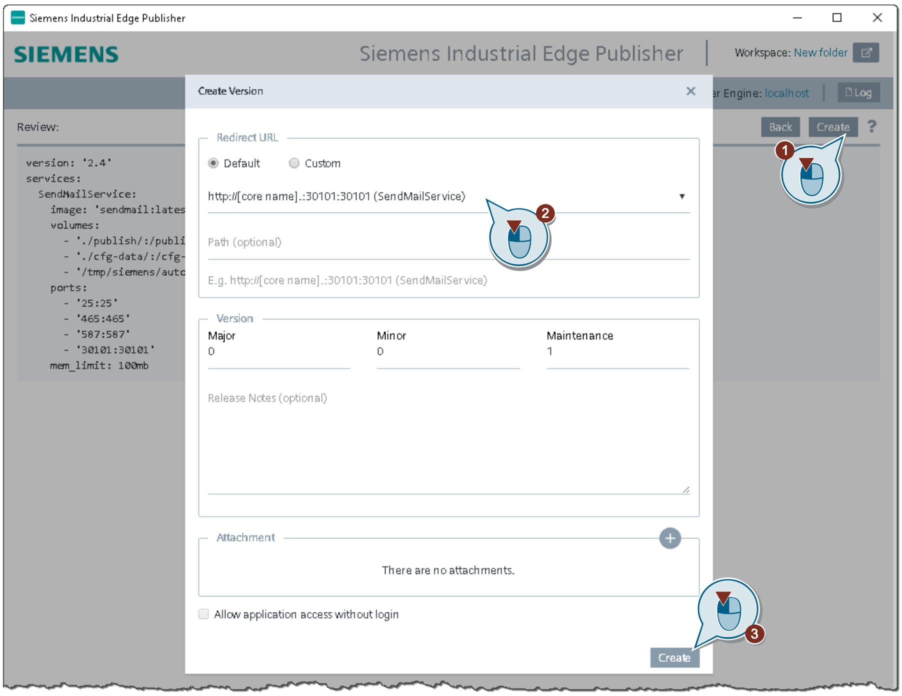
There are warnings on this page. Are you sure you want to continue?

```
sendmail:
  image: 'sendmail:latest'
  volumes:
    - './publish/:/publish/'
    - './cfg-data/:/cfg-data/'
    - '/tmp/siemens/automation:/tempcontainer/'
    - '/media/simatic:/media/simatic:ro,slave'
  ports:
    - '25:25'
    - '465:465'
    - '587:587'
    - '30101:30101'
  mem_limit: 100mb
```

Warning : Use external port (587) between range30000-35000 or otherwise it could conflict with the internal components on host side.
Warning : Use external port (465) between range30000-35000 or otherwise it could conflict with the internal components on host side.
Warning : Use external port (25) between range30000-35000 or otherwise it could conflict with the internal components on host side.

18. Create version and redirect

- First click "Create" (1).
- As needed, you can set the redirect URL via the dropdown field in this step if the application has to be reachable via web server. In this case, the web server must be reachable at port 30101 (2).
- Then click "Create" (3).



```
version: '2.4'
services:
  SendMailService:
    image: 'sendmail:lates
    volumes:
      - './publish/:/publi
      - './cfg-data/:/cfg-
      - '/tmp/siemens/autc
    ports:
      - '25:25'
      - '465:465'
      - '587:587'
      - '30101:30101'
    mem_limit: 100mb
```

● Default ○ Custom

http://[core name].:30101:30101 (SendMailService) ▾

Path (optional)

E.g. http://[core name].:30101:30101 (SendMailService)

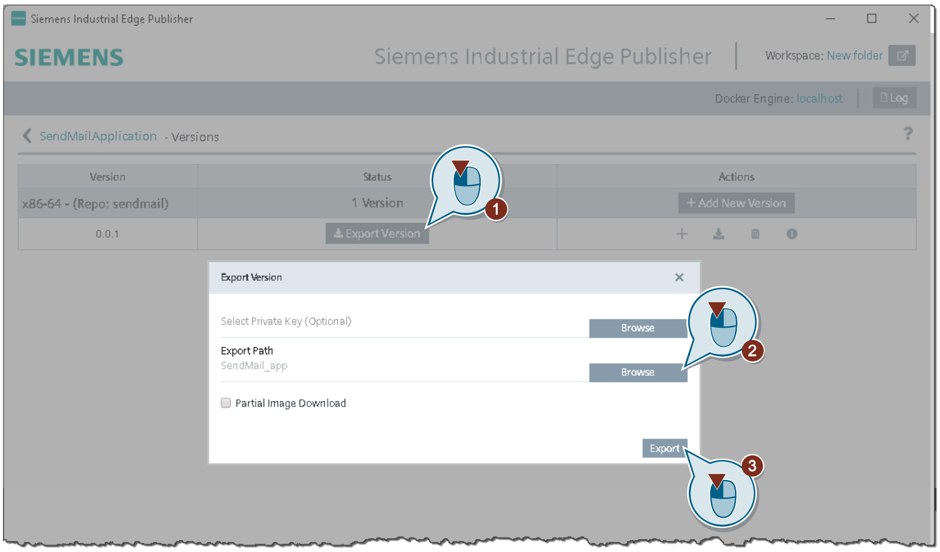
Major	Minor	Maintenance
0	0	1

Release Notes (optional)

There are no attachments.

□ Allow application access without login

19. Export the app
    - Click "Export version" (1).
    - Select a folder where you want to export the app (2).
    - Click "Create" (3)



Export Version

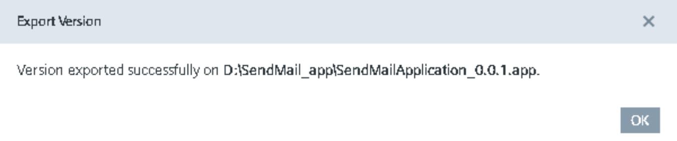
Select Private Key (Optional) Browse

Export Path: SendMail_app Browse

□ Partial Image Download

Export

Note The following message will appear when the app has exported successfully:



Export Version

Version exported successfully on D:\SendMail_app\SendMailApplication_0.0.1.app.

OK

In the "SendMail_app" folder you can now use "SendMailApplication_0.0.1.app" and transfer it to your SIMATIC HMI Unified Comfort Panel.

2.4 Manage: Manage edge app on the Unified Comfort Panel

This chapter describes:

1. Which TIA project settings are required for an edge app (see chapter 2.4.1) and
2. How you can transfer the exported app to your HMI device (see chapter 2.4.2).

2.4.1 Configuring TIA Portal project for edge apps

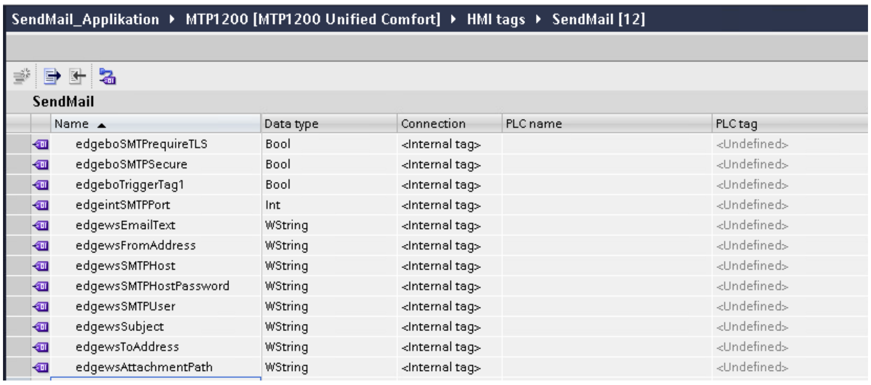
The following steps are required in TIA Portal in order to operate your project with an edge app.

Note Alternatively, you can also use the preconfigured example project. Download the file "Beispielprojekt.zip" ["Exampleproject.zip"] from this application example and extract it. You can then load it to your HMI device with TIA Portal.

Creating your own project

1. Create a TIA Portal project with a SIMATIC HMI Unified Comfort Panel.

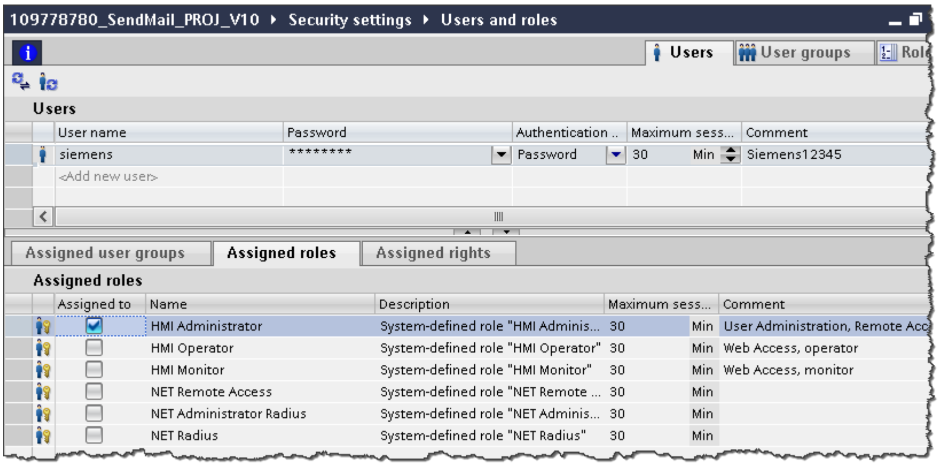
2. Create a tag table with the appropriate tag names and data types that you are also using in the edge app (see also chapter 2.2.2).


SendMail_Applikation ▸ MTP1200 [MTP1200 Unified Comfort] ▸ HMI tags ▸ SendMail [12]
SendMail

Name ▲	Data type	Connection	PLC name	PLC tag
edgeboSMTPrequireTLS	Bool	<Internal tag>		<Undefined>
edgeboSMTPSecure	Bool	<Internal tag>		<Undefined>
edgeboTriggerTag1	Bool	<Internal tag>		<Undefined>
edgeintSMTPPort	Int	<Internal tag>		<Undefined>
edgewsEmailText	WString	<Internal tag>		<Undefined>
edgewsFromAddress	WString	<Internal tag>		<Undefined>
edgewsSMTPHost	WString	<Internal tag>		<Undefined>
edgewsSMTPHostPassword	WString	<Internal tag>		<Undefined>
edgewsSMTPUser	WString	<Internal tag>		<Undefined>
edgewsSubject	WString	<Internal tag>		<Undefined>
edgewsToAddress	WString	<Internal tag>		<Undefined>
edgewsAttachmentPath	WString	<Internal tag>		<Undefined>

3. In your SIMATIC HMI Unified Comfort Panel, create a corresponding visualization for the email function and link up the pertinent tags.

4. Add a user with the role "HMI Administrator" to the User administration.


109778780_SendMail_PROJ_V10 ▸ Security settings ▸ Users and roles
Users
User groups
Rol
Users

User name	Password	Authentication ..	Maximum sess...	Comment
siemens	********	Password	30 Min	Siemens12345
<Add new user>				

Assigned to	Name	Description	Maximum sess...	Comment
☑	HMI Administrator	System-defined role "HMI Adminis...	30 Min	User Administration, Remote Acc
☐	HMI Operator	System-defined role "HMI Operator"	30 Min	Web Access, operator
☐	HMI Monitor	System-defined role "HMI Monitor"	30 Min	Web Access, monitor
☐	NET Remote Access	System-defined role "NET Remote ...	30 Min	
☐	NET Administrator Radius	System-defined role "NET Adminis...	30 Min	
☐	NET Radius	System-defined role "NET Radius"	30 Min	

Note Using the logon credentials for this user, you can log on to the local edge management of the SIMATIC HMI Unified Comfort Panel after the project is downloaded.

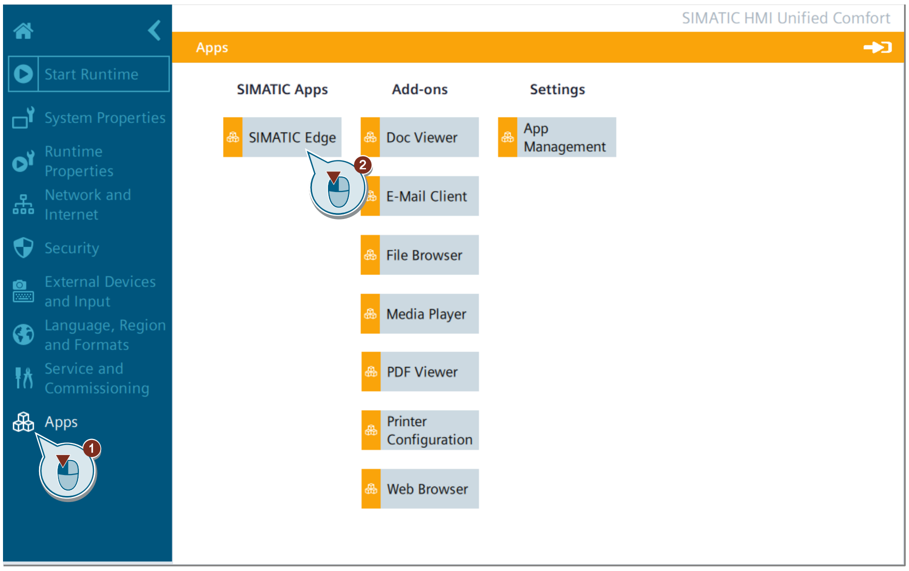
5. Save your project and download it to the SIMATIC HMI Unified Comfort Panel.

2.4.2 Installing edge app on the HMI device

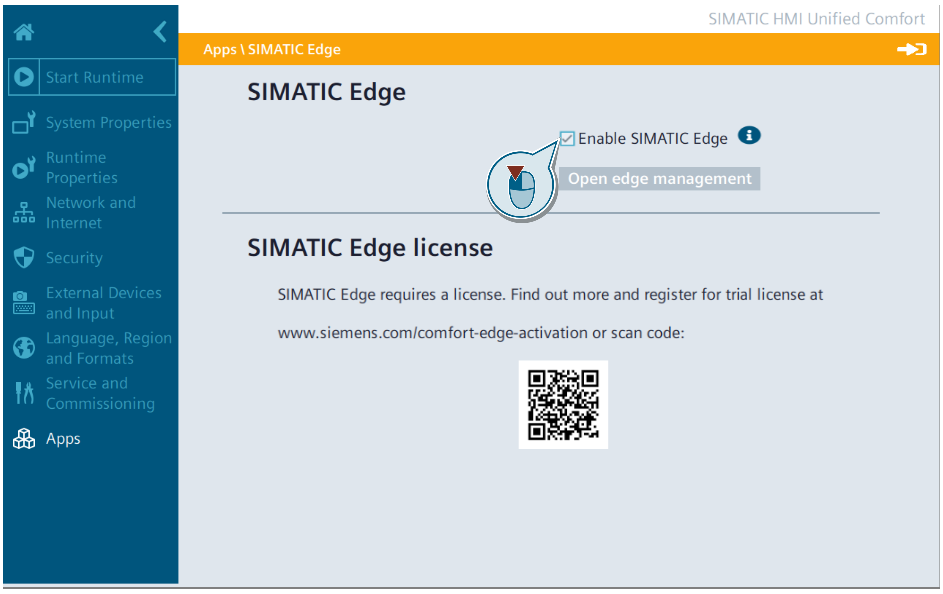
This chapter describes how you can enable the edge functionality on the operator device and install the edge app "SendMail" on the SIMATIC HMI Unified Comfort Panel.

Activating SIMATIC Edge

1. Open SIMATIC Edge
   - Start the SIMATIC HMI Unified Comfort Panel and open the "Apps" category in the Control Panel (1).
   - Then click "SIMATIC Edge" (2).



2. Enable the function "Enable SIMATIC Edge".



SIMATIC Edge

SIMATIC Edge license

SIMATIC Edge requires a license. Find out more and register for trial license at

www.siemens.com/comfort-edge-activation or scan code:

Note It may take a few seconds to enable SIMATIC Edge. When the activation is complete, the "Open edge management" button will appear dark gray.

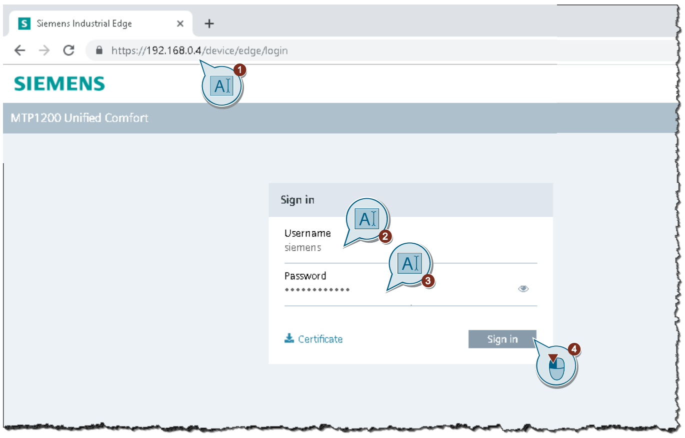
Transferring edge app

You have two ways of transferring the "SendMail" app to your operator device:

1. Transfer the app to the SIMATIC HMI Unified Comfort Panel on a USB drive and install it there via Edge Management.
2. Connect Edge Management via the network in order to install the edge app directly from your PC (see description below).

Open Edge Management via network and install app

1. Log on to Edge management
   - On your PC, open Chrome and enter "https://[IP address]" In this example, "https://192.168.0.4" (1)
   - Then enter "User name" (2) and "Password" (3) for the user from your TIA Portal project (see chapter 2.4.1).
   - Then click on "Sign in" (4).



https://192.168.0.4/device/edge/login

Username: siemens

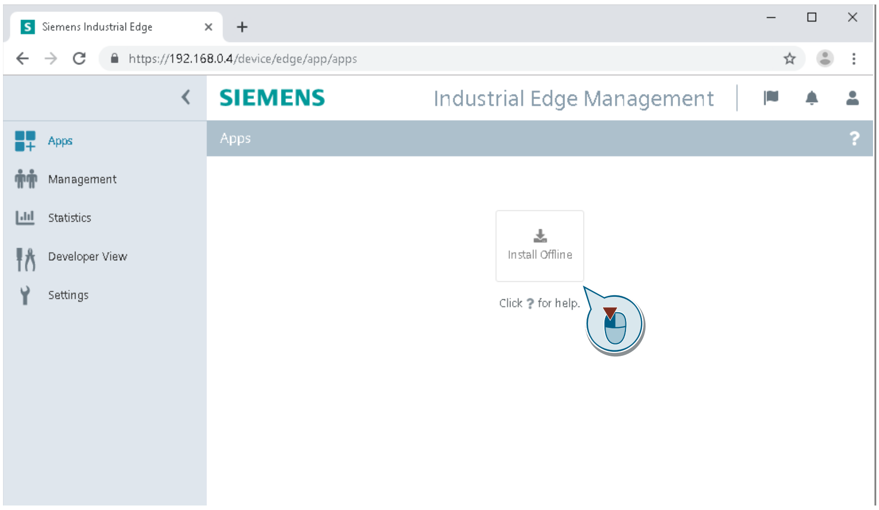
Password: ••••••••••••

Note

If you use the example project, you can log on with the following user credentials for the configured user.

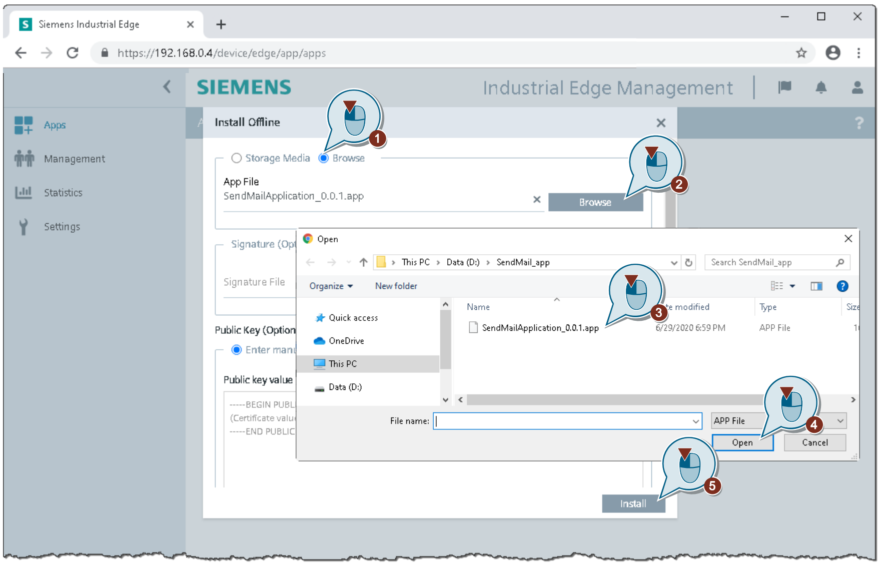
Username: siemens
Password: Siemens12345

2. In the Industrial Edge Management section of the Unified Comfort Panel, click "Install offline".



https://192.168.0.4/device/edge/app/apps

3. Select App
   - Select the "Browse" option (1) to browse your PC.
   - Then click "Browse" (2).
   - Select the exported edge app (3) and click "Open" (4).
   - Then click "Install" (5).

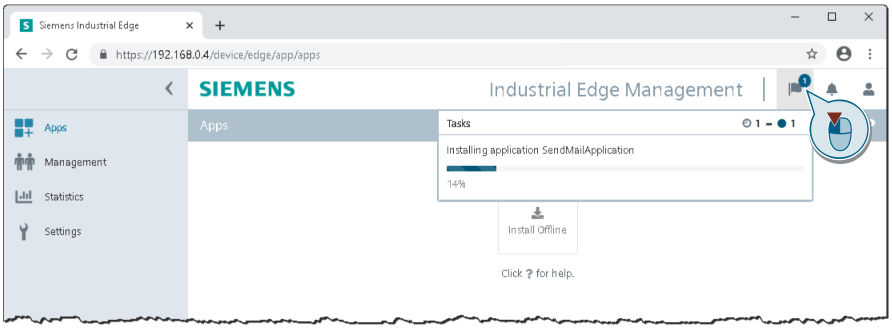


App File
SendMailApplication_0.0.1.app

-----BEGIN PUBL
(Certificate valu
-----END PUBLIC

SendMailApplication_0.0.1.app 6/29/2020 6:59 PM APP File

4. If you click the "Task" icon, you can monitor the current installation progress.

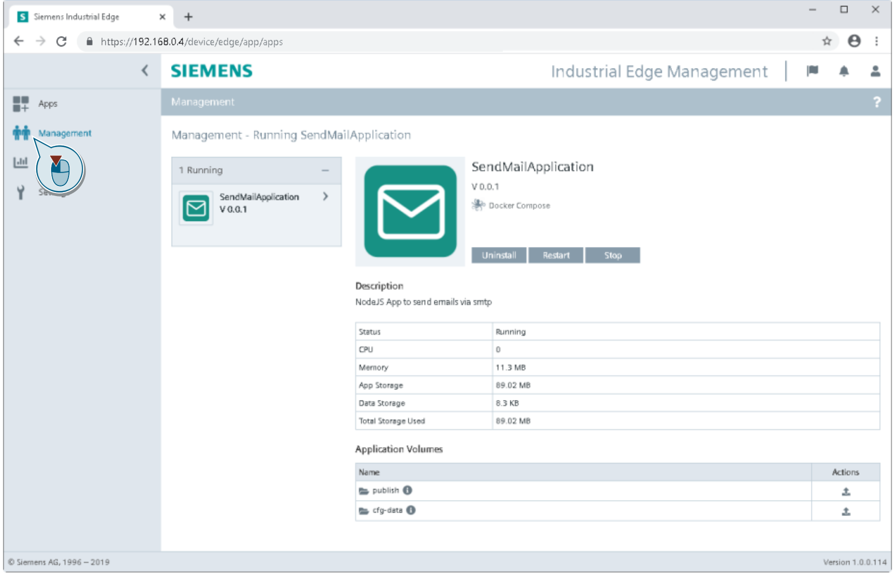


Installing application SendMailApplication

14%

Once the "SendMail" app has been successfully installed, the icon appears on the Edge Management start page.

5. If you then select the "Management" menu item, all installed edge apps and their statuses (e.g. "Running") will be displayed.



Management - Running SendMailApplication

SendMailApplication
V 0.0.1

Description
NodeJS App to send emails via smtp

Status	Running
CPU	0
Memory	11.3 MB
App Storage	89.02 MB
Data Storage	8.3 KB
Total Storage Used	89.02 MB

Application Volumes

Name	Actions
publish	⭱
cfg-data	⭱

3 Operating the example project

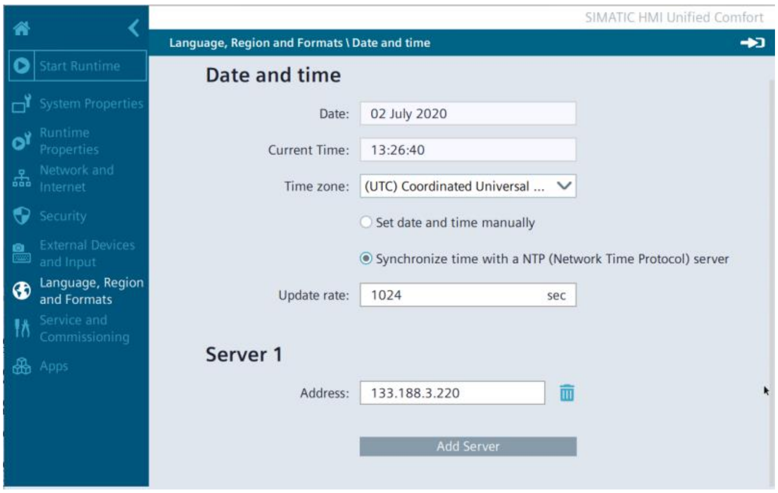
This chapter describes the essential function from the TIA Portal example project.

3.1 Unified Comfort Panel settings

Time stamp

In order to be able to use the SendMail edge app, you still have to set the current time on the Unified Comfort Panel or configure an NTP server.

Figure 3-1


SIMATIC HMI Unified Comfort
Language, Region and Formats \ Date and time
Start Runtime
System Properties
Runtime Properties
Network and Internet
Security
External Devices and Input
Language, Region and Formats
Service and Commissioning
Apps

Date and time

Date: 02 July 2020

Current Time: 13:26:40

Time zone: (UTC) Coordinated Universal ...

○ Set date and time manually

● Synchronize time with a NTP (Network Time Protocol) server

Update rate: 1024 sec

Server 1

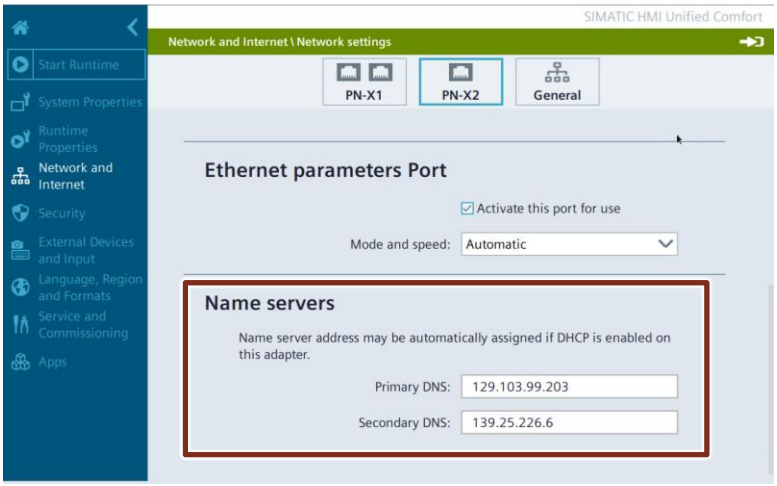
Address: 133.188.3.220

Add Server

DHCP on X2

The X2 network interface must also be correctly configured. In the event that you have set DHCP for this interface, then you also have to enter the DNS server.

Figure 3-2


SIMATIC HMI Unified Comfort
Network and Internet \ Network settings
PN-X1
PN-X2
General
Start Runtime
System Properties
Runtime Properties
Network and Internet
Security
External Devices and Input
Language, Region and Formats
Service and Commissioning
Apps

Ethernet parameters Port

☑ Activate this port for use

Mode and speed: Automatic

Name servers

Name server address may be automatically assigned if DHCP is enabled on this adapter.

Primary DNS: 129.103.99.203

Secondary DNS: 139.25.226.6

3.2 Editing SMTP settings

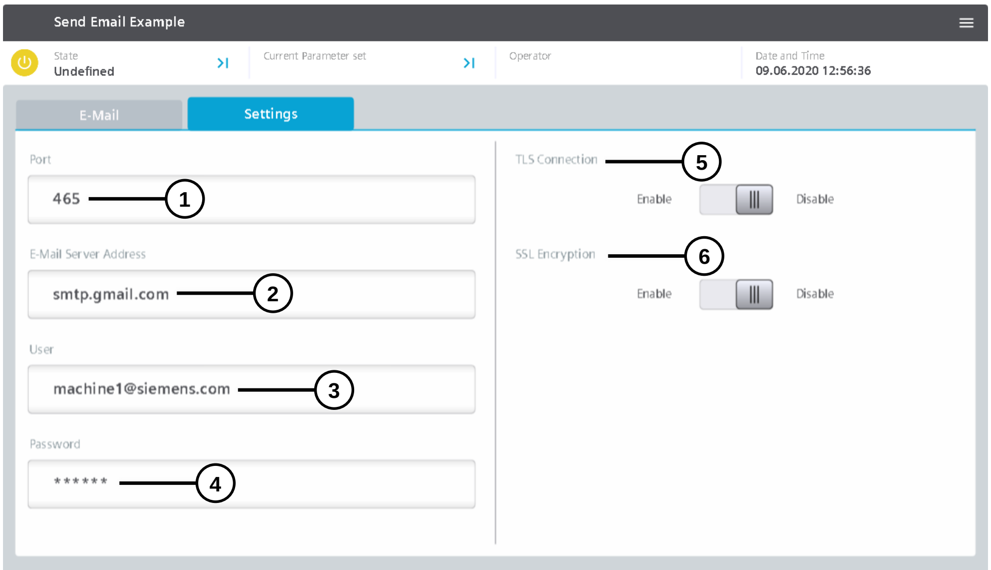
If the runtime and the SendMail app have been started on the Unified Comfort Panel, the settings for the SMTP server still have to be made before it is possible to send email. The application can connect to most SMTP servers.

Note

> Please refer to the documentation from your email provider for the connection parameters of your email account.

Using the example of the Google Mail server, this means the following:

Figure 3-3



Send Email Example

State: Undefined

Current Parameter set

Operator

Date and Time: 09.06.2020 12:56:36

E-Mail | Settings

Port: 465

E-Mail Server Address: smtp.gmail.com

User: machine1@siemens.com

Password: ******

TLS Connection: Enable / Disable

SSL Encryption: Enable / Disable

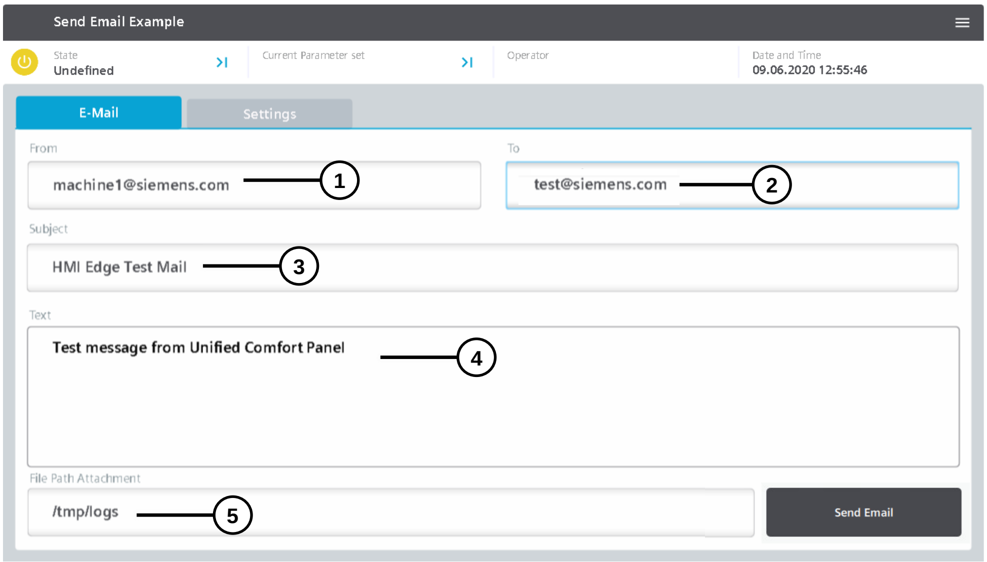
Table 3-1

No.	Description
①	Port of the SMTP server
②	Address of the server
③	User name
④	Password
⑤	Enable TLS encryption
⑥	Enable SSL encryption

3.3 Creating an email

You can then enter the details of the email, compose the text, and select the file if you wish to attach one.

Figure 3-4



Send Email Example

State: Undefined

Current Parameter set

Operator

Date and Time: 09.06.2020 12:55:46

From: machine1@siemens.com

To: test@siemens.com

Subject: HMI Edge Test Mail

Text: Test message from Unified Comfort Panel

File Path Attachment: /tmp/logs

Table 3-2

No.	Description
1	Email address of the sender
2	Email address of the recipient
3	Subject line
4	Email body
5	Path to the file to be attached

4 Useful information

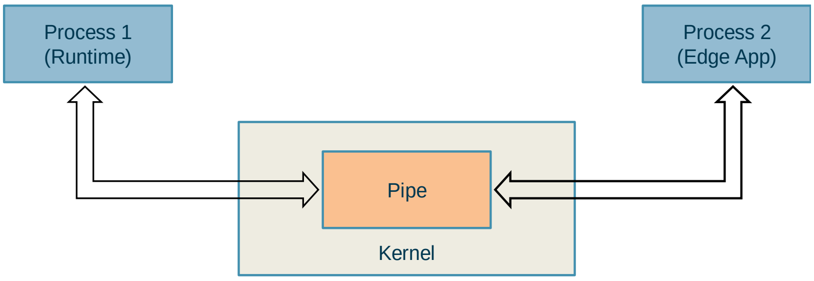
4.1 Named Pipe

A Pipe or Pipeline is a data stream between two processes through a buffer operating on the First In First Out (FIFO) principle. In plain terms, this means that the output of a computer program is used as the input for another program.

Note You can find additional information on the topic of Open Pipe in the operator's manual "SIMATIC WinCC Unified Open Pipe".

https://support.industry.siemens.com/cs/ww/en/view/109778823/134221616523

Figure 4-1

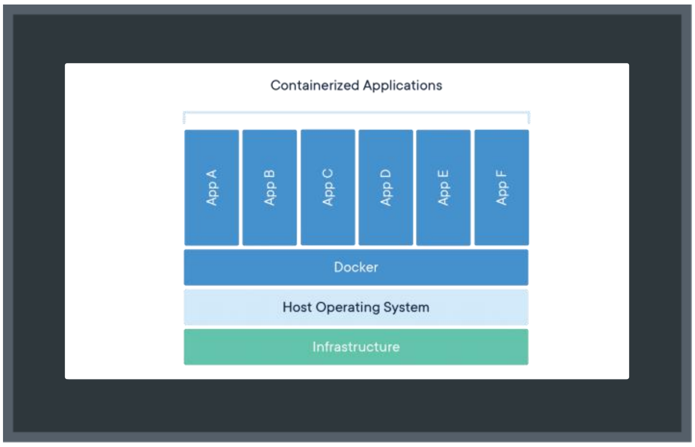


4.2 Docker

General

Docker is a free piece of software for isolating applications with container virtualization. Docker simplifies the deployment of applications because containers can be easily transferred and installed as files along with all necessary packages.

Figure 4-2



Container Service (Docker)

In order to create a container image, you will need an instruction list for how the container will be generated (Docker file), as well as a container engine in order to build the image. The Docker Engine can be installed in Windows or Linux. This application example refers to Docker in Windows.

4.3 SIMATIC Edge runtime

The SIMATIC Edge runtime makes it possible to run various services and applications in parallel with the Unified Comfort Panel runtime. To do this, an environment is established within the operating system of the Panels on which applications can be launched in container format. These are in the popular Docker format.

The container then runs in a separate part of the system, sandboxed away from the main part of the system in a secure manner. User-created programs can run in this container. You can create these applications in JavaScript or Python, for example.

5 Appendix

5.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.

Please send queries to Technical Support via Web form:

support.industry.siemens.com/cs/my/src

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

5.2 Industry Mall

The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:
mall.industry.siemens.com

5.3 Links and Literature

Table 5-1

No.	Subject
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the article page of the application example https://support.industry.siemens.com/cs/ww/en/view/109778780
\3\	Download – Siemens Industry Edge Publisher https://support.industry.siemens.com/cs/ww/en/view/109778875
\4\	Programming manual - Industrial Edge Publisher https://support.industry.siemens.com/cs/ww/en/view/109778824
\5\	Operating manual - SIMATIC HMI WinCC Unified Open Pipe https://support.industry.siemens.com/cs/ww/en/view/109778823
\6\	Manual "Unified Comfort Panels for Industrial Edge" https://support.industry.siemens.com/cs/ww/en/view/109773259

5.4 Change documentation

Table 5-2

Version	Date	Change
V1.0	07/2020	First version
V1.1	04/2021	Bug fixes