This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No 871481.

TRUSTS Trusted Secure Data Sharing Space

D3.1 Infrastructure set-up and technical operations

Document Summary Information

<table>
<thead>
<tr>
<th>Grant Agreement No</th>
<th>Acronym</th>
<th>TRUSTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>871481</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Full Title</th>
<th>TRUSTS Trusted Secure Data Sharing Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Date</td>
<td>01/01/2020</td>
</tr>
<tr>
<td>Duration</td>
<td>36 months</td>
</tr>
<tr>
<td>Project URL</td>
<td><a href="https://trusts-data.eu/">https://trusts-data.eu/</a></td>
</tr>
<tr>
<td>Deliverable</td>
<td>D3.1 Infrastructure set-up and technical operations</td>
</tr>
<tr>
<td>Work Package</td>
<td>WP3</td>
</tr>
<tr>
<td>Contractual due date</td>
<td>31/03/2020</td>
</tr>
<tr>
<td>Actual submission date</td>
<td>31/03/2020</td>
</tr>
<tr>
<td>Nature</td>
<td>Report Dissemination Level Public</td>
</tr>
<tr>
<td>Lead Beneficiary</td>
<td>LSTECH ESPANA</td>
</tr>
<tr>
<td>Responsible Author</td>
<td>Evangelos Kotsifakos, Rosa Araujo</td>
</tr>
<tr>
<td>Contributions from</td>
<td>FNET, eBOS, RELATIONAL RO, FORTH</td>
</tr>
</tbody>
</table>
Revision history (including peer reviewing & quality control)

<table>
<thead>
<tr>
<th>Version</th>
<th>Issue Date</th>
<th>% Complete</th>
<th>Changes</th>
<th>Contributor(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.0</td>
<td>10/02/20</td>
<td>5</td>
<td>Initial Deliverable Structure</td>
<td>Evangelos Kotsifakos, Rosa Araujo (LST)</td>
</tr>
<tr>
<td>V2.0</td>
<td>28/02/20</td>
<td>70</td>
<td>First draft</td>
<td>Evangelos Kotsifakos, Rosa Araujo (LST)</td>
</tr>
<tr>
<td>V3.0</td>
<td>22/03/20</td>
<td>90</td>
<td>Second draft, ready for first review</td>
<td>Marius Parashiv, Santiago Andrés Azcoitia, Rosa Araujo (LST)</td>
</tr>
<tr>
<td>V4.0</td>
<td>23/03/20</td>
<td>95</td>
<td>Peer review comments</td>
<td>Bin Li (FhG)</td>
</tr>
<tr>
<td>V5.0</td>
<td>30/03/20</td>
<td>100</td>
<td>Final version</td>
<td>Marius Parashiv, Santiago Andrés Azcoitia, Rosa Araujo (LST)</td>
</tr>
</tbody>
</table>

Disclaimer

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the TRUSTS consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the TRUSTS Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the TRUSTS Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© TRUSTS, 2020-2022. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.
Table of Contents

Document Summary Information....................................................................................................................... 1
Revision history (including peer reviewing & quality control) ................................................................. 2
Disclaimer.................................................................................................................................................. 2
Copyright message..................................................................................................................................... 2
Table of Contents......................................................................................................................................... 3
List of Figures............................................................................................................................................. 3
List of Tables............................................................................................................................................... 3
Glossary of terms and abbreviations used.................................................................................................... 4
1 Executive Summary.................................................................................................................................. 5
2 Introduction............................................................................................................................................ 5
  2.1 Mapping Projects’ Outputs.................................................................................................................. 6
  2.2 Deliverable Overview and Report Structure......................................................................................... 7
3 Environment............................................................................................................................................ 7
  3.1 Environment requirements................................................................................................................... 7
4 Technology selection............................................................................................................................... 7
5 Security and privacy................................................................................................................................ 9
6 Resources requirements.......................................................................................................................... 10
7 Environment realization......................................................................................................................... 10
  7.1.1 Environment access..................................................................................................................... 14
  7.1.2 Deployment.................................................................................................................................. 16
8 Conclusions and Next Actions................................................................................................................ 18

List of Figures

Figure 1: Containers and VMs [https://www.docker.com/resources/what-container]................................. 8
Figure 2: The development environment overview...................................................................................... 9
Figure 3: Google cloud administration dashboard....................................................................................... 12
Figure 4: Google cloud interface details .................................................................................................... 13
Figure 5: Google cloud service monitoring .................................................................................................. 14
Figure 6: Google cloud Compute engine – VM instance details................................................................. 15
Figure 7: Docker set-up............................................................................................................................... 16

List of Tables

Table 1: Adherence to TRUSTS GA Deliverable & Tasks Descriptions ....................................................... 6
D3.1. Infrastructure set-up and technical operations

### Glossary of terms and abbreviations used

<table>
<thead>
<tr>
<th>Abbreviation / Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>GA</td>
<td>Grant Agreement</td>
</tr>
<tr>
<td>GB</td>
<td>Gigabyte</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>MB</td>
<td>Megabyte</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>VCS</td>
<td>Version Control System</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
</tbody>
</table>
1 Executive Summary

This deliverable provides an overview of the infrastructure and guidelines on how to manage the environment set up in the TRUSTS context, so that the development team can use the most crucial services as soon as possible. The goal is to have a development environment ready from the beginning of the projects in order for the technical partners to be able to initiate their development. The current design is an initial version, which will be continuously improved and updated in future deliverable in month 12 (December 2020).

A Google Cloud development environment has been set up that allows developers to securely access it and use Dockers to implement and deploy initial versions of their applications. A Virtual Machine has been allocated for this purpose and these resources will be extended in the future according to the development needs.

Dockers have been chosen for application isolation, security, extensibility and efficiency.

The details are described in this deliverable. The environment is managed by LSTech.

2 Introduction

This deliverable is related to the task T3.1 - Infrastructure set-up and technical operations [M1-M36]. The task’s goal is to provide a stable and secure environment for developing and hosting the project’s components.

The TRUSTS platform provides a set of capabilities that enable operators to develop applications with a high degree of privacy by design capabilities.

For this environment, LSTech will employ DevOps and all the state-of-the-art mechanisms to support the architecture and specifications defined in T2.4 “Architecture design and technical specifications (D2.4 months 12 and 24).

The administration and security aspects will be the main concern for this task. LSTech provides a dev-ops environment that will allow the continuous integration, automated unit testing and collaboration, employing tools like Kubernetes\(^2\) and Docker\(^3\) for deployment, Jenkins\(^4\) for continuous integration, and Jupyter\(^5\) and GitHub\(^6\) for collaboration.

The task makes sure that components of the TRUSTS platform, that implement the APIs specified in T2.4, for the integration of external data-sets and platforms can be deployed, updated and maintained easily in production and test environments.

The aim of this task is initially to provide a quick start environment for the development of the trusts platform components. Although, the technical specifications will be continuously adapted according to new requirements coming up, from the outcome of task 2.4 and during the integration and deployment phases. In this deliverable we provide an overview on the services of the infrastructure and guidelines on how to use it in

1 \(\text{https://cloud.google.com/}\)
2 \(\text{https://kubernetes.io/}\)
3 \(\text{https://www.docker.com/}\)
4 \(\text{https://jenkins.io/}\)
5 \(\text{https://jupyter.org/}\)
6 \(\text{https://github.com/}\)
2.1 Mapping Projects' Outputs

Purpose of this section is to map TRUSTS Grant Agreement commitments, both within the formal Deliverable and Task description, against the project’s respective outputs and work performed.

### Table 1: Adherence to TRUSTS GA Deliverable & Tasks Descriptions

<table>
<thead>
<tr>
<th>Trusts Platform implementation</th>
<th>Respective Document Chapter(s)</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3.1 Infrastructure set-up and technical operations</td>
<td>A stable and secure environment for developing and hosting the project’s components will be offered by LSTech. LSTech will employ DevOps and all the state-of-the-art mechanisms to support the architecture and specifications defined in T2.4. Setup, administration and security will be the main concern for this task. LSTech will provide a dev-ops, continuous integration, automated unit testing and collaboration environment, employing tools like Kubernetes and Docker for deployment, Jenkins for continuous integration, and Jupyter and Github for collaboration. The preferred development methodology is Agile, supported by online collaboration tools for product development. The task makes sure that components of the TRUSTS platform, that implement the APIs specified in T2.4, for the integration of external data-sets and platforms can be deployed, updated and maintained easily in production and test environments</td>
<td>Sections 3 - 7</td>
</tr>
</tbody>
</table>

TRUSTS Infrastructure Deliverable

**D3.1 TRUTS Infrastructure**

The deliverables consist of documents that provide an overview on the services of the infrastructure and guidelines how to use it in the TRUSTS context. An initial version should be available in M3 so that development can use the most crucial services as soon as possible. The infrastructure will be continuously improved with an updated report in M12.
2.2 Deliverable Overview and Report Structure

The structure for this Deliverable technical is the following.

We describe the environment technical requirements in section 3 and in section 4 we describe the technology selection for the environment. Security and privacy issues are mentioned in section 5 while in section 6 we describe the resources used. In section 7 we provide the details of the environment realization, how to access it and how to deploy applications. Last section is the Conclusions.

3 Environment

3.1 Environment requirements

In this section we describe the basic requirements for the development environment of a cloud platform applicable for TRUSTS. After analysing the Description of Action of the TRUSTS project we conclude that the development platform for TRUSTS should support the following general requirements:

- **Availability**
  The platform shall be available 24/7 in order for the partners to be able to develop, deploy and test their applications.

- **Continuous integration**
  The TRUSTS project follows a lean methodology, where elements of the architecture and infrastructure will evolve based on the continuous feedback from developers. In order to allow rapid yet safe evolution of systems, an automated build, test and deploy process will be provided. DevOps tools will allow the developers to build and test their applications directly on the cloud.

- **Security**
  The environment should be secure with restricted access and no connection to the external world. If it is necessary for applications to connect to external applications, this will be done through secure procedures.

- **Extensibility**
  The environment should be easily extensible. New applications should be easily integrated.

- **Interoperability**
  The environment should allow connections to external services when needed.

- **Administration and manageability**
  The environment should allow easy administration and management.

4 Technology selection

LSTech is providing a cloud-based environment for the infrastructure set-up and technical operations, using Google Cloud. Google infrastructure and servers are robust and tools are provided to ensure data security with backup, monitoring and encryption also available.

---

7 [https://cloud.google.com/](https://cloud.google.com/)
D3.1. Infrastructure set-up and technical operations

The related security and privacy assurances can be overviewed here https://cloud.google.com/security/overview/

Google is compliant with the European law and it has high security standards that are adequate for the TRUSTS platform.

For the development of the TRUSTS platform we are using a Dockerized environment. This will allow ease of implementation, extensibility, portability and security.

A container [https://www.docker.com/resources/what-container] is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Docker containers that run on Docker Engine:

- Standard: Docker created the industry standard for containers, so they could be portable anywhere
- Lightweight: Containers share the machine’s OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs
- Secure: Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry.

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.[https://www.docker.com/resources/what-container]

![Figure 1: Containers and VMs](https://www.docker.com/resources/what-container)
Each TRUSTS partner will create their own Docker application that will be hosted in a VM of the infrastructure. The following figure illustrates this setup.

![Diagram of development environment](image)

**Figure 2: The development environment overview**

### 5 Security and privacy

Applications and programming interfaces (APIs) will be designed, developed, deployed, and tested in accordance with leading industry standards and adhere to applicable legal, statutory, or regulatory compliance obligations.

The environment will be accessible through only one secure point using SSH and HTTPS protocols. The development environment will be accessible only to specific people related to the technical work packages and their applications.

The development environment will be accessible only to specific people related to the technical WPs and their applications. We will be daily monitoring the infrastructure to identify possible inconsistencies, conflicts or security issues based on system, application and access logs.

Backups will be taken periodically, once every week in the beginning of the project. As the project advances we will examine the possibility of more frequent backups. LSTech’s team will be responsible for restoring the environment in case of failure in 48 hours.
All the servers will be hosted in EU countries and no data will be transferred to other countries outside the EU.

In case special agreements are needed between the partners to ensure secure access to the developed software and to disclosure of information or data, these will be drafted and signed according to the counselling of the project’s legal partners and we will implement the appropriate measures.

The technical partners may also provide additional special security and privacy requirements for their components.

6 Resources requirements

Initially, a shared Virtual Machine will be used for all TRUSTS technical partners. If and when needed a multi-VM configuration (Kubernetes) will be deployed.

The resources that are currently allocated are the following:

One Virtual Machine serving as development environment with characteristics:

- Machine type: Memory optimized
- Virtual CPUs: 2
- Ram: 13GB
- Disc size: 80G
- Image OS: Debian 10 - Buster

Services:
- Docker: 19.3.07
- Docker-compose: 1.25.4
- Docker network: Trustsnet

The resources are adequate to support the initial development and they will be updated properly according to the needs of the development.

7 Environment realization

The infrastructure has been set-up on LSTech Google Cloud, and policies and procedures shall be established and maintained in support of data security to include confidentiality, integrity, and availability across multiple system interfaces, jurisdictions, and business functions to prevent improper disclosure, alteration, or destruction.

The measures taken in order to put in place the security protocols and measures for this type of environment have been:
D3.1. Infrastructure set-up and technical operations

- Unique identification and authentication: Internal corporate user account credential shall be restricted for ensuring appropriate identity, entitlement, and access management and in accordance with established policies and procedures:
  - One account per user on Google Cloud
  - One account per user on Gitlab.com

- Version Control Systems
  
  Git is a distributed Version Control System (VCS) where each local code repository maintains a complete copy of the history of changes to code.
  
  A TRUSTS group on gitlab.com will be registered.

- Package Repositories
  
  A different private repository for each component/application will be set up.

  The technical partners will be asked to provide a README.md file for describing application functionality and deployment instructions. It should be noted that each partner will deploy their application.

  If the partner’s code shall remain private, they will have available the use the Gitlab\(^8\) repository for your compiled application, instead of the source code.

  This infrastructure is realized in the Google Cloud provided by LSTech. Dashboards and tools to manage the infrastructure are available by Google and in the following screenshots you may see examples of the management user interface. The dashboard will be accessible only by LSTech administrators.

\(^8\) [http://Gitlab.com](http://Gitlab.com)
D3.1. Infrastructure set-up and technical operations

Figure 3: Google cloud administration dashboard
### D3.1. Infrastructure set-up and technical operations

![Google cloud interface details](image)

**Figure 4: Google cloud interface details**
7.1.1 Environment access

For accessing the infrastructure, there will be one account per user on Google Cloud.
For the development there will be also one account per user on Gitlab.com

Each partner will be responsible for their application’s deployment, on the shared Docker instance.
Docker files should be used, and Docker-compose is to be preferably implemented.

Version Control System

LSTech will provide for the developing environment:

- One TRUSTS group on gitlab.com.
- One different private repository for each component/application.
- README.md file for describing application functionality and deployment instructions.

If the code should remain private, partners are asked to use the Gitlab repository for their compiled application, instead of the source code.

Initially, 1 shared VM will be used for all partners. If and when needed a multi-VM configuration (probably with Kubernetes) will be deployed.
D3.1. Infrastructure set-up and technical operations

Each partner will be responsible for their application’s deployment, on the shared Docker instance.
Docker-files should be used, and Docker-compose is to be implemented.

```
$ docker version
Client: Docker Engine - Community
  Version:      19.03.7
  API version: '1.40'
  Go version:  go1.12.17
  Git commit:  5141c95a4f
  Built:       Wed Mar 4 01:22:41 2020
  OS/Arch:     linux/amd64
  Experimental: false

Server: Docker Engine - Community
  Version:      19.03.7
  API version: '1.40' (minimum version 1.12)
  Go version:  go1.12.17
  Git commit:  5141c95a4f
  Built:       Wed Mar 4 01:21:19 2020
  OS/Arch:     linux/amd64
  Experimental: false
```

Figure 7: Docker set-up

### 7.1.2 Deployment

Instructions for TRUSTS technical partners:

- Users login to LSTech Google cloud.
- Pull their code from their Git repository (including the Docker-file and Docker-compose file), build the Docker image and deploy it.
- If a partner doesn’t want to share their code, they can share compiled versions of it.

**GITLAB.COM project access and users**

- The Group TRUSTS Platform has been created. https://gitlab.com/groups/trusts-platform
- Each project member will be invited and they will create their code repository for version control

**Deployment on Google Cloud**

**Setting up GCloud (first time):**

- Install Google Cloud SDK (Instructions available here: Windows, Linux, MacOS).
- Open Google Cloud SDK Shell (or just the terminal on mac/linux) and run gcloud init.
  - Login with your Google credentials
  - If asked, click yes to generate the required ssh keys.
- You might need to close and re-open the terminal.
Connecting to GCloud:
Run the Gcloud compute command to connect:
`Gcloud beta compute --project "trustseu" ssh --zone "europe-west1-b" "trusts-development"`

Deploying on GCloud:
The Docker daemon is running and you have been given access.
To check if it works for you, just run “Docker run hello-world”.

Suggestion for deployment:
The easiest way would be to have all the necessary files in your Git repo (including a Docker file and Docker-compose), and to pull it in the VM when there are changes. You can then easily run “Docker-compose restart” to apply the changes.

A Docker network has been created called “trusts net”.
Please add your container to the TRUSTS net network (documentation).
Example Docker-compose.yml using trusts net network:
```
version: '2'
services:
  nginx:
    container_name: nginx
    image: nginx:latest
    networks:
      default:
        external:
          name: trustsnet
```

Contact and Help
Collaboration environment
For an easy and immediate communication between the different partners and the developers a SLACK\(^9\) channel is available.

https://trusts-dev.slack.com/

Technical Support
For assisting any question or technical issues about the development infrastructure, we provide direct technical support. Details can be found in the online documentation:

https://docs.google.com/document/d/1PARgGNsFkml1hMQr6YEjN-xdqKQfh5MozSkJffj2PM/edit?usp=sharing

\(^9\) https://slack.com/
8 Conclusions and Next Actions

In order for the TRUSTS partners to be able to initiate the development of their applications, we have set up a development environment in Google Cloud. This document provides the details of the development environment realization for the TRUSTS project. The technologies and the resources available as well as instructions for accessing it and deploying applications are described and they are also available in an online shared document that will be updated regularly. The environment is managed by LSTech and it will be updated to meet the development needs of the project. A second updated version of this document will be delivered at the end of 2020.