TRUSTS Trusted Secure Data Sharing Space

D3.2 TRUSTS Infrastructure II

Document Summary Information

Grant Agreement No	871481	Acronym	TRUSTS
Full Title	TRUSTS Trusted Secure Data Sharing Space		
Start Date	01/01/2020	Duration	36 months
Project URL	https://trusts-data.eu/		
Deliverable	D3.2 TRUSTS Infrastructure II		
Work Package	WP3		
Contractual due date	31/12/2020	Actual submission date	16/12/2020
Nature	Report	Dissemination Level	Public
Lead Beneficiary	LSTECH ESPANA		
Responsible Author	Evangelos Kotsifakos, Rosa Araujo, Xavier Olivares		
Contributions from	FNET, eBOS, RELATIONAL RO, FORTH		

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

Revision history (including peer reviewing & quality control)

Version	Issue Date	% Complete	Changes	Contributor(s)
v1.0	1/10/2020	40%	initial draft, D3.2 revision	Rosa Araujo, Evangelos Kotsifakos, Xavier Olivares (LST)
V2.0	1/11/2020	60%	draft restructuring and update	Rosa Araujo, Evangelos Kotsifakos, Xavier Olivares (LST)
V3.0	1/12/2020	70%	added information, comments from partners	Rosa Araujo, Evangelos Kotsifakos, Xavier Olivares (LST)
V4.0	10/12/2020	80%	completed first draft for review	Rosa Araujo, Evangelos Kotsifakos, Xavier Olivares (LST)
V5.0	14/12/2020	100%	incorporated reviewers' comments/ ready for submission	Benjamin Heitmann (FhG), Evangelos Kotsifakos, Xavier Olivares Rosa Araujo (LST)

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 however 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

Table of Contents

List of Figures

List of Tables

Glossary of terms and abbreviations used

Abbreviation / Term	Description
API	Application Programming Interface
GA	Grant Agreement
GB	Gigabyte
HTTP	Hypertext Transfer Protocol
MB	Megabyte
OS	Operating System
VCS	Version Control System
VM	Virtual Machine
SDK	Software Development Kit
SSH	Secure Shell

1 Executive Summary

The objective of this deliverable is to provide a description of the infrastructure and guidelines on how to manage the environment to be used in the TRUSTS project. The scope of the TRUSTS infrastructure is to have a development environment ready from the beginning of the project for the technical partners to be able to run their developments. The platform provides a set of features that enable operators to develop applications with a high degree of privacy by design capabilities.

The current design is an update of the initial version drafted in Deliverable D3.1 and submitted in March 2020. The most important differences with respect to the previous deliverable are:

- The inclusion of potential new tools to be used for having a better control of the infrastructure as the project progresses (section 4)
- Detail of the required information for accessing the environment (section 5)
- User management and roles (section 7)
- Infrastructure size limitations (section 7.1.3)

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 TRUSTS's components.

2 Introduction

This document details the implementation plan for resource allocation for TRUSTS infrastructure, which considers CPUs, GPUs, RAM, and storage requirements. To satisfy the requirements we are using Google Cloud which has a compute cluster with 3 compute nodes providing 6 CPUs, 21 GB RAM and 3 storage nodes. There are 3 snapshots scheduled to back up the main storage data node daily.

For this environment, LSTech is employing DevOps and all the state-of-the-art mechanisms to support the development of the components from the partners and meet the needs for the architecture and specifications defined in T2.4 “Architecture design and technical specifications” (D2.4 months 12 and 24).

The administration and security aspects are the main concern of this task. LSTech provides a DevOps environment that will allow the continuous integration, automated unit testing and collaboration, employing tools like Kubernetes[1] and Docker[2] for deployment, Jenkins[3] for continuous integration, and Jupyter[4] and GitHub[5] for collaboration.

The task 3.1 “Infrastructure set-up and technical operations”, 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.

[1] https://kubernetes.io/

[2] https://www.docker.com/

[3] https://jenkins.io/

[4] https://jupyter.org/

[5] https://github.com/

The aim of this task is to provide a site for collaboration and development of the TRUSTS platform components. The technical specifications will be continuously adapted until the end of the project, according to new requirements coming up from the outcome of task 2.4 and during the integration and deployment phases.

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

Trusts Platform implementation		**Respective Document Chapter(s)**	**Justification**
T3.1 Infrastructure set-up and technical operations	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	Sections 3 - 7	Section 3: Environment Section 4: Technology selection Section 5: Security and privacy Section 6: Resources requirements Section 7: Environment realization
TRUSTS Infrastructure Deliverable			
D3.2 TRUSTS Infrastructure The deliverable provides an overview on the services of the infrastructure and guidelines on how to use it in the TRUSTS context. This is the second and last related deliverable regarding the development infrastructure.			

2.2 Deliverable Overview and Report Structure

The structure for this technical deliverable is the following.

The document starts by a description of the environment requirements in section 3, while in section 4, a description of the selected technology is provided. Security and privacy issues are mentioned in section 5. In section 6 a description of the resources used is explained. Section 7 comprises the details of the environment realization, how to access it, how to deploy applications and also our size limitations. Finally, the last section is devoted to 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.

The infrastructure must be a dynamic, powerful, and robust centralised environment, able to provide TRUSTS members a system to access and share their different components without affecting each other, within their own workspace.

To this end, each technical partner will have at least one user to manage their workspace, with the proper access rights.

4 Technology selection

LSTech is providing a cloud - based environment for the infrastructure set-up and technical operations, using **Google Cloud**[6]. Google infrastructure and servers are robust, and tools are provided to ensure data security with backup, monitoring and encryption also available.

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**[7] 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

[6] https://cloud.google.com/

[7] https://www.docker.com/resources/what-container

isolated processes in user space. Containers take up less space than Virtual Machines (VM), (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.

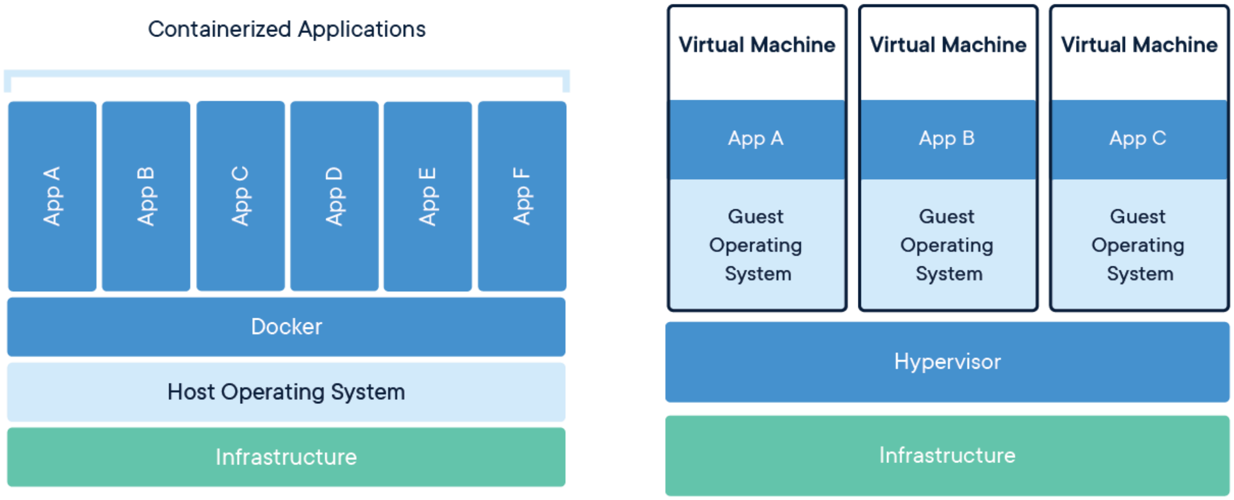


Figure 1: Containers and VMs

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.

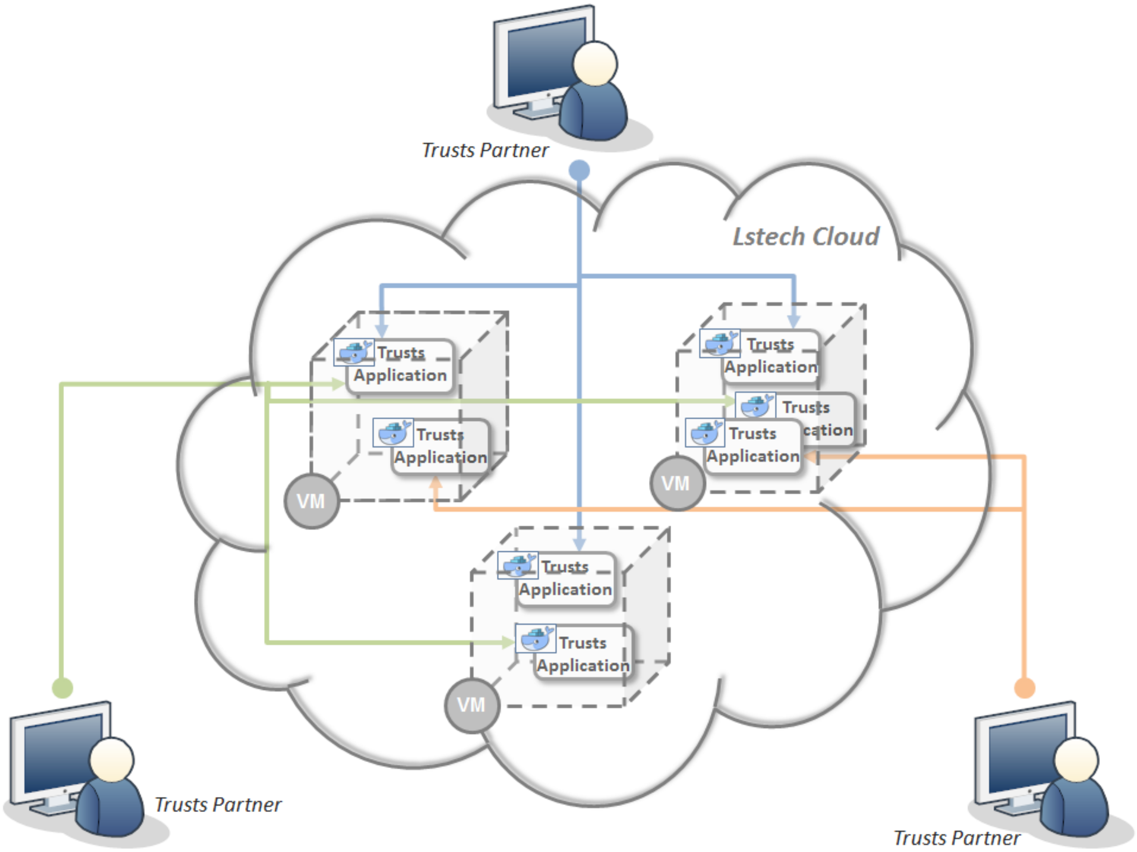


Figure 2: The development environment overview

According to the size progression of the project, new tools will be used in order to have better control of the infrastructure. Orchestration tools such as Ansible[8] or Puppet[9] will be set up as the number of Virtual Machines keep growing. Both technologies are open source and have a freemium version. Kubernetes will also be set up for the container management and deployment.

5 Security and privacy

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

So that the administrator of the site can grant access to the shared space, previously the developers have to fill in an excel file with the relevant information to manage their permissions and allocate the needed resources. The required information is the following:

- ❏ Partner name
- ❏ Module/component
- ❏ Contact person
- ❏ Email of the contact person
- ❏ Users to access The GCloud/ Gmail account
- ❏ Gitlab email
- ❏ Dependences
- ❏ Main technology/framework used
- ❏ Docker images of the software
- ❏ Docker compose or swarm
- ❏ Minimum requirements (CPU/RAM/disk)

On a daily basis the infrastructure is monitored to identify possible inconsistencies, conflicts or security issues based on system, application and access logs.

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.

Backups are taken periodically, once every week in the beginning of the project. As the project advances more frequent backups will be implemented. LSTech’s team will be responsible for restoring the environment in case of a failure within 48 hours.

All the servers will be hosted in EU countries and no data will be transferred to other countries outside the EU.

[8] https://www.ansible.com/

[9] https://puppet.com/

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 is to be used for all TRUSTS technical partners. Following the criterion of on demand service, when necessary a multi-VM configuration (Kubernetes) will be deployed.

The resources currently allocated are the following:

Three virtual machines: One generic with more RAM and two are on-demand for a particular partner. All of them serving as development environment:

The main VM has the following specifications:

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 2 extra virtual machines have the following specifications:

Machine type: Standard Medium performance

Virtual CPUs: 2

Ram: 4GB

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 current developments and they will be updated properly according to the needs of the progression of the different components.

7 Environment realization

The infrastructure has been set-up on LSTech Google Cloud, and policies and procedures are being 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 actions taken in order to put in place the security protocols and measures for this type of environment are:

- **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 has been registered.

- **Package Repositories**

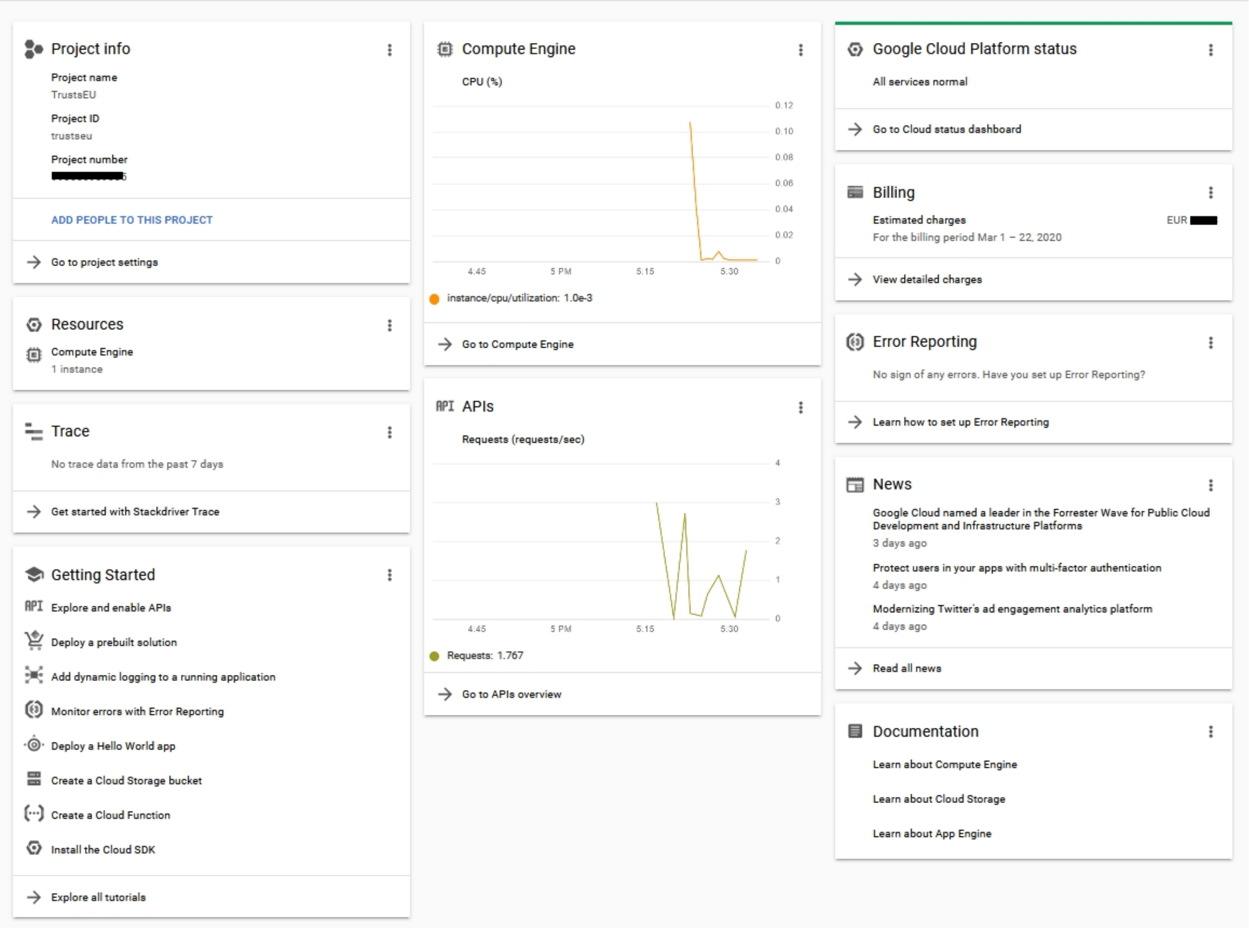
  A different private repository for each component/application will be set up.

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 should be kept private, the developers will have available the use the Gitlab[10] repository for their compiled application, instead of the source code.

Dashboards and tools to manage the infrastructure are available by Google and in the following screenshots we provide examples of the management user interface. The dashboard is accessible only by LSTech administrators.

[10] http://Gitlab.com



Project info

Project name
TrustsEU

Project ID
trustseu

Project number
[redacted]

ADD PEOPLE TO THIS PROJECT

→ Go to project settings

Resources

Compute Engine
1 instance

Trace

No trace data from the past 7 days

→ Get started with Stackdriver Trace

Getting Started

Explore and enable APIs

Deploy a prebuilt solution

Add dynamic logging to a running application

Monitor errors with Error Reporting

Deploy a Hello World app

Create a Cloud Storage bucket

Create a Cloud Function

Install the Cloud SDK

→ Explore all tutorials

Compute Engine

CPU (%)

● instance/cpu/utilization: 1.0e-3

→ Go to Compute Engine

APIs

Requests (requests/sec)

● Requests: 1.767

→ Go to APIs overview

Google Cloud Platform status

All services normal

→ Go to Cloud status dashboard

Billing

Estimated charges EUR [redacted]
For the billing period Mar 1 – 22, 2020

→ View detailed charges

Error Reporting

No sign of any errors. Have you set up Error Reporting?

→ Learn how to set up Error Reporting

News

Google Cloud named a leader in the Forrester Wave for Public Cloud Development and Infrastructure Platforms
3 days ago

Protect users in your apps with multi-factor authentication
4 days ago

Modernizing Twitter's ad engagement analytics platform
4 days ago

→ Read all news

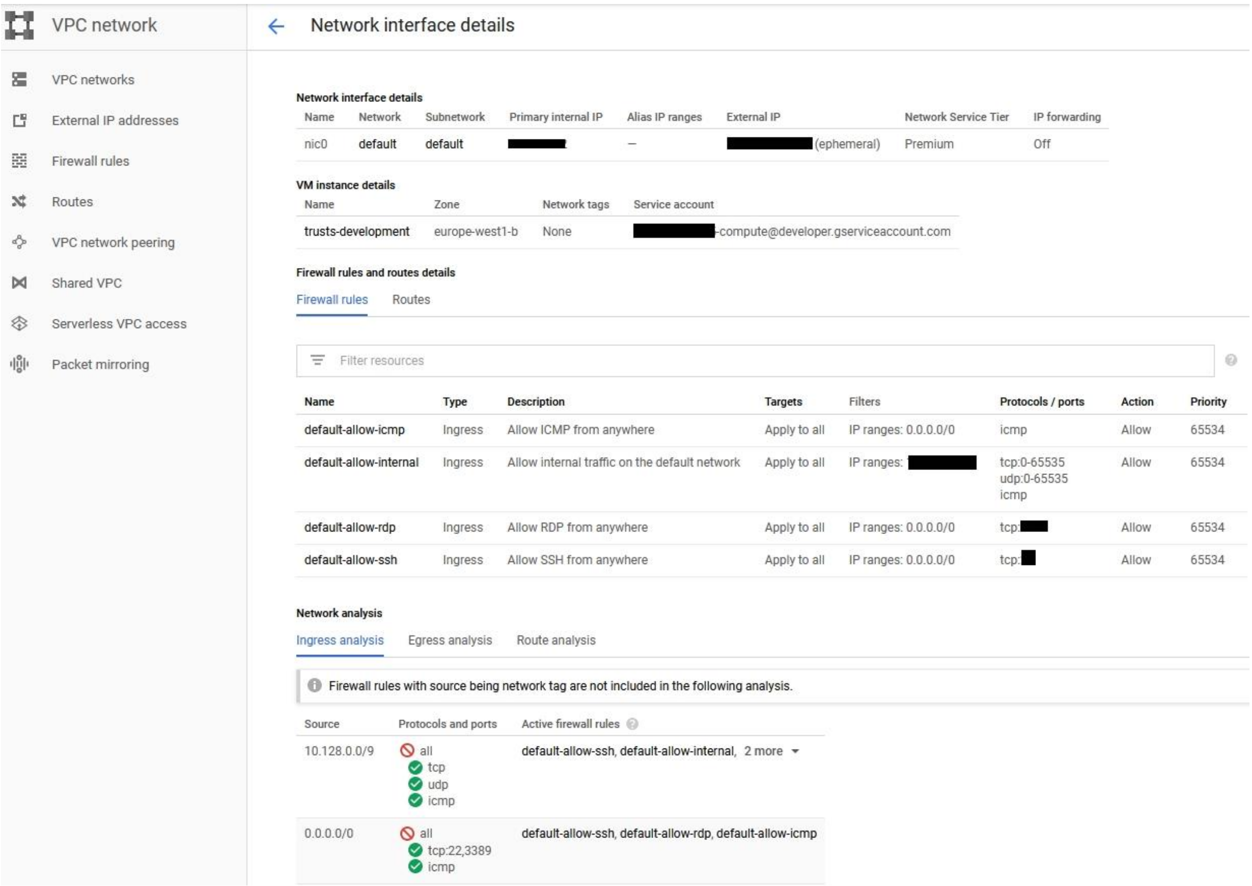
Documentation

Learn about Compute Engine

Learn about Cloud Storage

Learn about App Engine

Figure 3: Google cloud administration dashboard



Network interface details

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Service Tier	IP forwarding
nic0	default	default	[redacted]	–	[redacted] (ephemeral)	Premium	Off

VM instance details

Name	Zone	Network tags	Service account
trusts-development	europe-west1-b	None	[redacted]-compute@developer.gserviceaccount.com

Firewall rules and routes details

Firewall rules   Routes

Filter resources

Name	Type	Description	Targets	Filters	Protocols / ports	Action	Priority
default-allow-icmp	Ingress	Allow ICMP from anywhere	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534
default-allow-internal	Ingress	Allow internal traffic on the default network	Apply to all	IP ranges: [redacted]	tcp:0-65535 udp:0-65535 icmp	Allow	65534
default-allow-rdp	Ingress	Allow RDP from anywhere	Apply to all	IP ranges: 0.0.0.0/0	tcp:[redacted]	Allow	65534
default-allow-ssh	Ingress	Allow SSH from anywhere	Apply to all	IP ranges: 0.0.0.0/0	tcp:[redacted]	Allow	65534

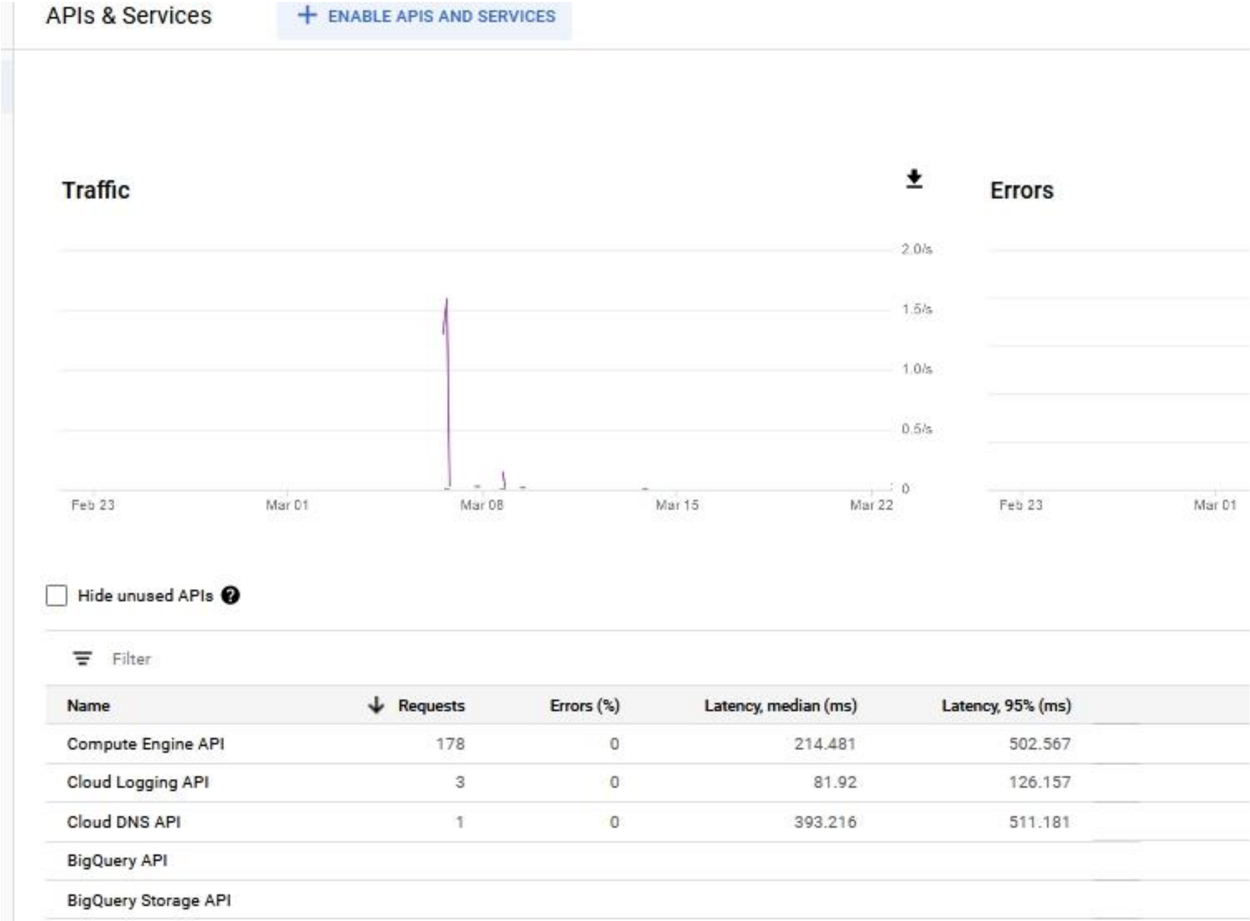
Network analysis

Ingress analysis   Egress analysis   Route analysis

Firewall rules with source being network tag are not included in the following analysis.

Source	Protocols and ports	Active firewall rules
10.128.0.0/9	⊘ all ✓ tcp ✓ udp ✓ icmp	default-allow-ssh, default-allow-internal, 2 more ▾
0.0.0.0/0	⊘ all ✓ tcp:22,3389 ✓ icmp	default-allow-ssh, default-allow-rdp, default-allow-icmp

Figure 4: Google cloud interface details



□ Hide unused APIs

Name	↓ Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
Compute Engine API	178	0	214.481	502.567
Cloud Logging API	3	0	81.92	126.157
Cloud DNS API	1	0	393.216	511.181
BigQuery API				
BigQuery Storage API				

Figure 5: Google cloud service monitoring

7.1.1 Environment access

For accessing the infrastructure, one account per user on the Google Cloud has been set-up.

For the development there is also one account per user on Gitlab.com

Each partner is responsible for their application's deployment, on the shared Docker instance.

Docker files should be used, and Docker-compose is to be preferably implemented.

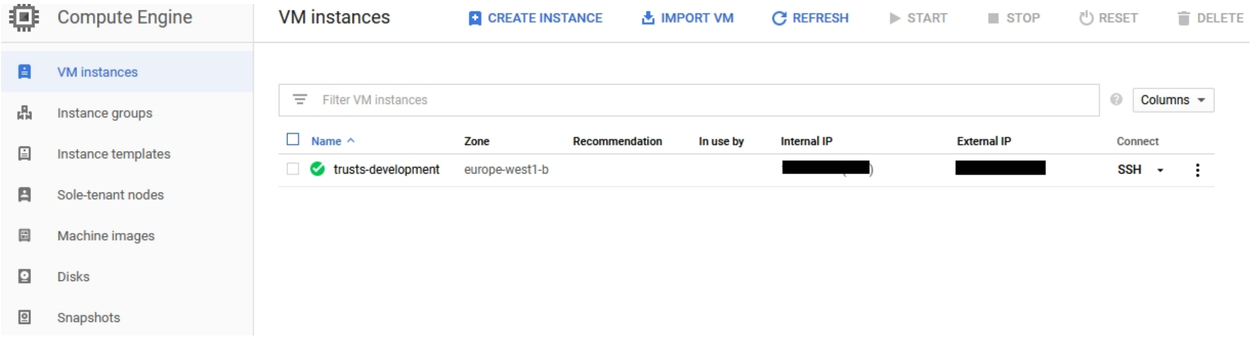
Version Control System

LSTech provides 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, a shared VM will be used for all partners. If and when needed a multi-VM configuration (probably with Kubernetes) will be deployed.



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
trusts-development	europe-west1-b			[redacted]	[redacted]	SSH

Figure 6: Google cloud Compute engine – VM instances

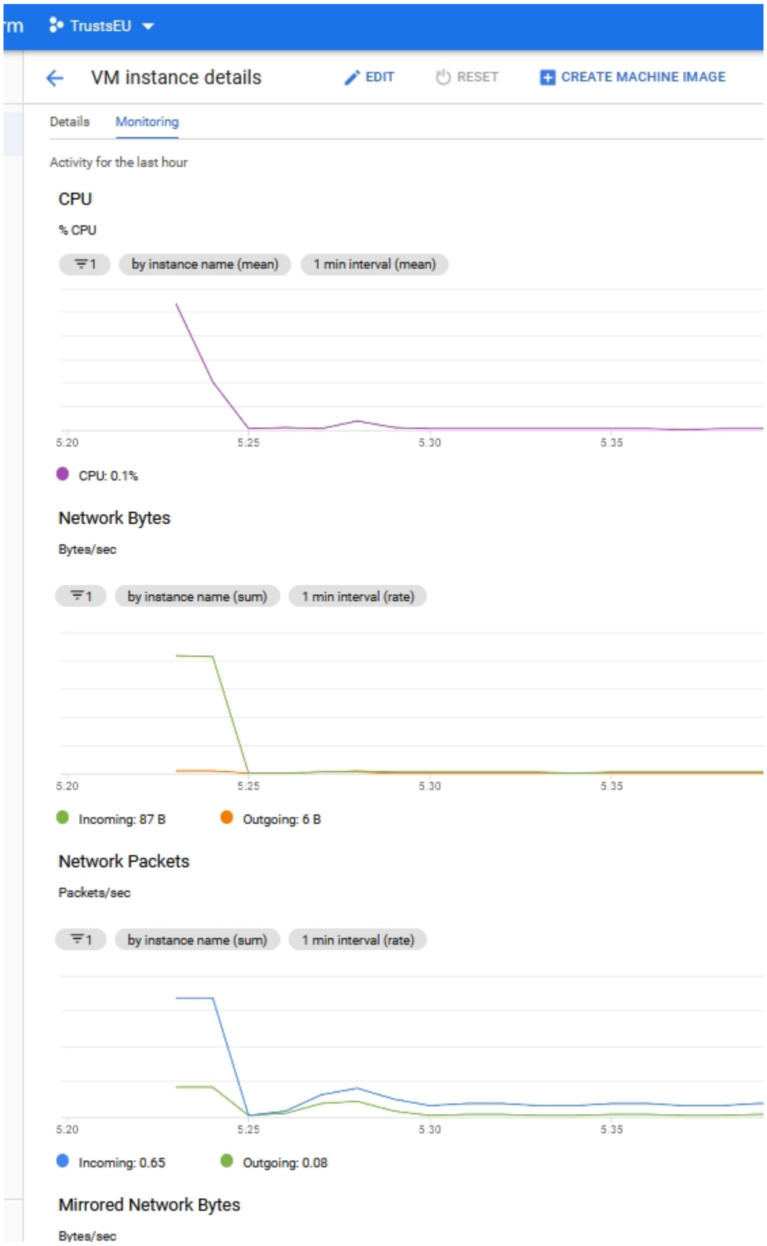
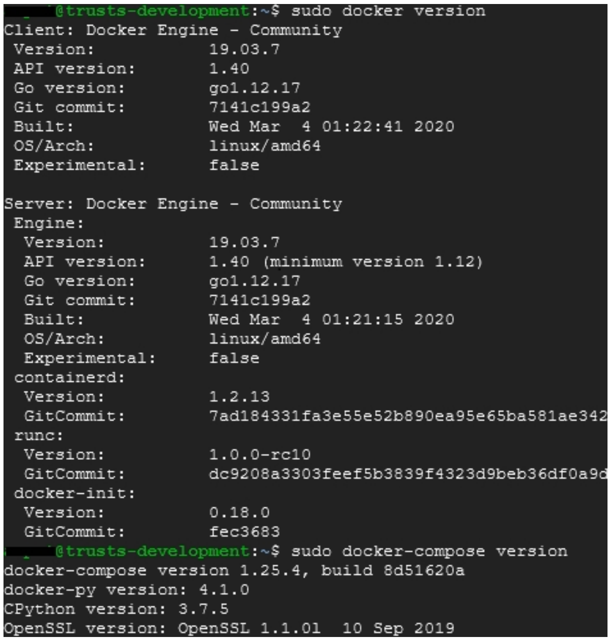


Figure 7: Google cloud Compute engine – VM instances details



```
[redacted]@trusts-development:~$ sudo docker version
Client: Docker Engine - Community
 Version:           19.03.7
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        7141c199a2
 Built:             Wed Mar  4 01:22:41 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.7
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.17
  Git commit:       7141c199a2
  Built:            Wed Mar  4 01:21:15 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.2.13
  GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae342
 runc:
  Version:          1.0.0-rc10
  GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9d
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
[redacted]@trusts-development:~$ sudo docker-compose version
docker-compose version 1.25.4, build 8d51620a
docker-py version: 4.1.0
CPython version: 3.7.5
OpenSSL version: OpenSSL 1.1.0l  10 Sep 2019
```

Figure 8: Docker set-up

7.1.2 Deployment

User management and roles:

The Google Cloud IAM is the component that allows users to have certain permissions when accessing the infrastructure. Currently, the roles assigned to all the partners are: Compute OS Login and Service Account User. These are GCloud premade roles that fit correctly to our current use case.

Compute Os Login role allows:

- Get and Log in into all the deployed VM instances
- Get the Resource manager projects
- Check the quotas and service usage benchmarks and costs.

Service Account User role allows:

- Check the IAM roles for the partners
- Get the Resource manager projects

A custom role will be created for the purpose of having a clearer control and management of the user roles, since there is some permission overlap between the current two roles. This way will be easier to modify the accesses in the big picture.

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:

- 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.
- Partners might need to close and re-open the terminal.

Connecting to GCloud:

From Gcloud UI:

Access the Google cloud portal: https://console.cloud.google.com/

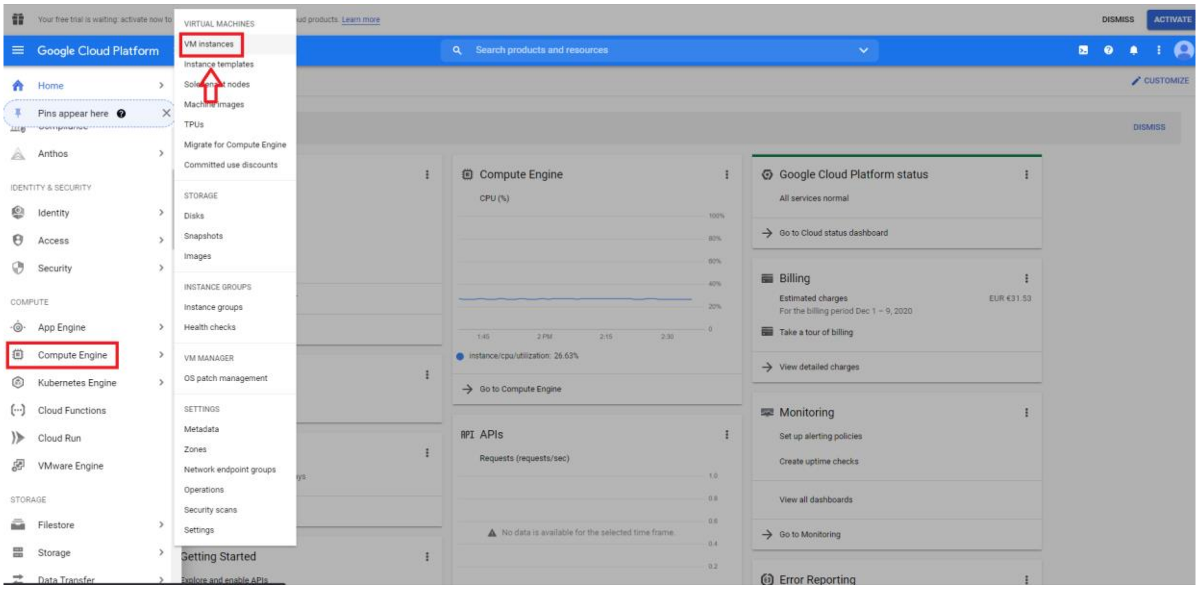
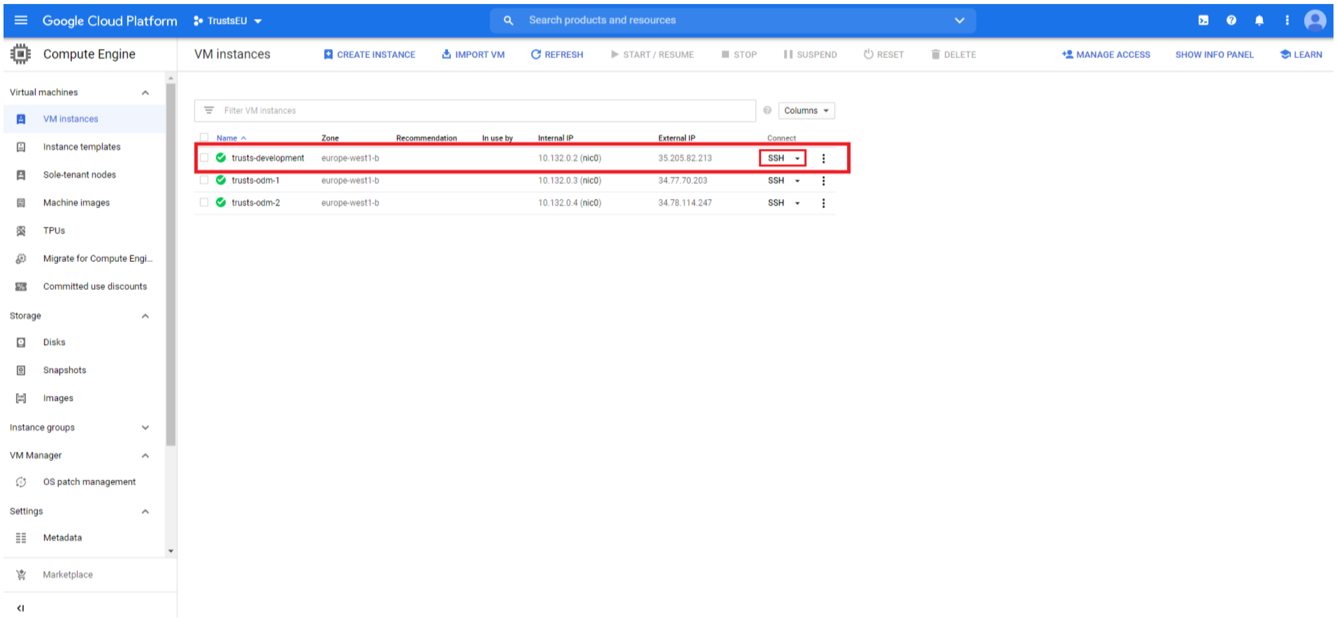


Figure 9: GCloud UI main menu

To SSH into the general usage VM:



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
trusts-development	europe-west1-b			10.132.0.2 (nic0)	35.205.82.213	SSH
trusts-odm-1	europe-west1-b			10.132.0.3 (nic0)	34.77.70.203	SSH
trusts-odm-2	europe-west1-b			10.132.0.4 (nic0)	34.78.114.247	SSH

Figure 10: Gcloud VM instances list

There are many ways to log in into the VM. The easiest way to do it is just pressing into the SSH and letting the browser open a local terminal.

It is also possible to generate SSH keys and log in from the Terminal.

Instructions given to the developers:

From Terminal:

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 partners have been given access.

To check if it works , 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. Partners can then easily run “Docker-compose restart” to apply the changes.

A Docker network has been created called “trusts net”.

To add partners’ containers 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

A SLACK[11] workplace has been selected and is available to all technical members and developers for easy and immediate communication.

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

Access to the SLACK workplace and its channels is done through a related request to LSTech.

Technical Support

For supporting the project's development and resolving technical issues about the development infrastructure, communication channels have been opened providing direct technical support. Details can be found in the online documentation:

https://docs.google.com/document/d/1PARgGNsFkmI1hMQr6YEjN-WxdqKQfh5MozSkJffj2PM/edit?usp=sharing

7.1.3 Infrastructure size limitations

In terms of the type and amount of resources, we have used a tool provided by Google, Google Cloud Pricing Calculator, which allows us to estimate costs per resource in an approximate way, being able to add multiple variables in order to increase reliability and accuracy.

With the current budget, the maximum amount of resources that can be provided are:

Using the e2-standard-2 type of instance (2 vCPUs, 8GB Memory), we can provide 12 running simultaneously, 24/7. Each machine will have a hard drive and if requested, a static domain.

Depending on the geographical situation in which the machines are deployed, the price varies a lot, we have used Munich as an intermediate point, but there are 6 different cities within the range of possibilities in Western Europe. If the geographical situation is within North America, we would be able to extend the size to 15 VMs at the same cost. This is not an option since we prefer our servers to reside within the EU for privacy reasons.

[11] https://slack.com/

8 Conclusions and Next Actions

With the purpose of offering the TRUSTS partners a site to facilitate the development of their applications, a training environment in Google Cloud has been set up. This document provides an update of the technical details for the 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 is being updated on a regular basis.