Author	Kaspar Schiess (Pryv SA)
Reviewer	Ilia Kebets (Pryv SA)
Version	6 (20181022)

Pryv IO Deployment

Design Guide, Sizing and Operational Concerns

Introduction

You're reading the Deployment Design Guide for the Pryv IO middleware. This document will guide you through the process of deciding which hardware/virtual machines to provision to host your Pryv IO instance. It will help you decide on sizing and give you hints on how to operate your Pryv cluster.

This document describes how to deploy Pryv IO 1.3. For more information about Pryv and Pryv IO releases, please refer to the Pryv website and in particular our 'News' section.

Architecture

Overview

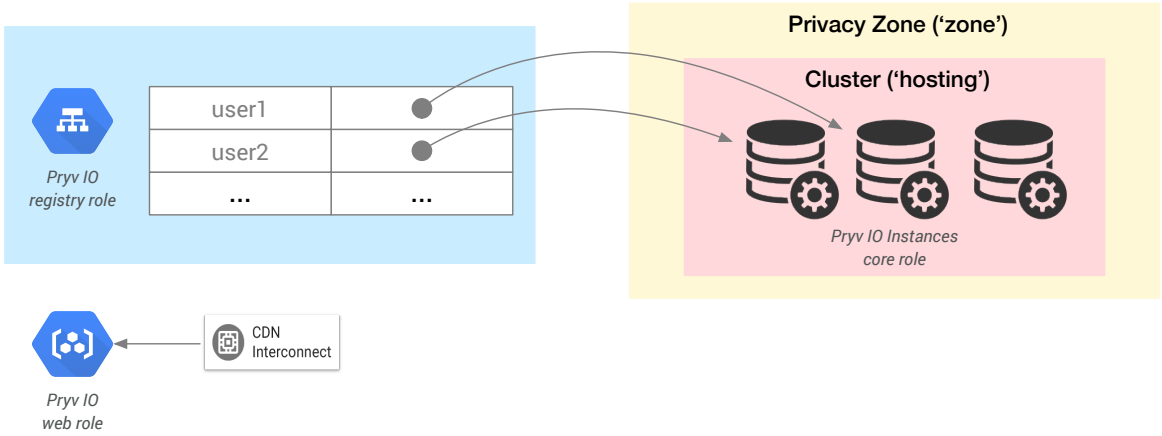
Any Pryv IO deployment is composed of three separate functions: A web server serving static files, called the 'web' role in this document. Secondly, a service discovery component, called the 'registry' role in this document. And finally, a component that stores data and regulates access to the data, called the 'core' role.

Normal deployments will only have one server in the *web* role. This server will mostly serve static files and can be put behind a CDN like cloudfront for large deployments. The *web* role is optional: when the application allows it, its function can be delegated to hosts serving static sites like Github.

The *registry* servers associate users (usernames) to the servers that store data. They are a partition directory and are responsible for routing requests. This function is achieved via HTTPS and also via a DNS interface. We recommend to dedicate at least two servers to the *registry* role, configured in a leader/follower configuration.

The information stored here is not (very) sensitive - the only thing an attacker can learn from these systems is that a user exists in a Pryv IO installation.

The *core* servers store the user data. As such, they are the most privacy-sensitive components of the system. *Core* servers are partitioned by user name. A single server can either store all the data from an installation or - in the other extreme - the data of a single user. As you will see in this document, there are two primary reasons for partitioning the *core* role: Privacy and Load. Both reasons influence your deployment and should guide your decisions.

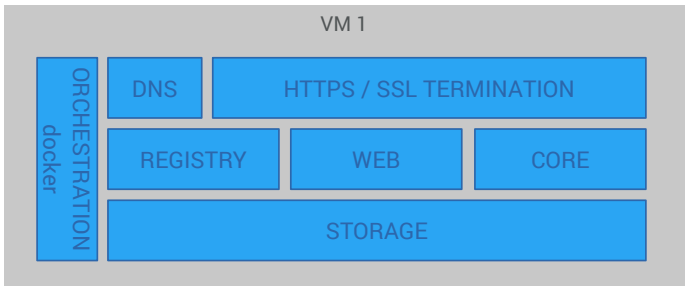


The diagram above shows a Pryv IO installation from an logical point of view.

Deployment Scenarios

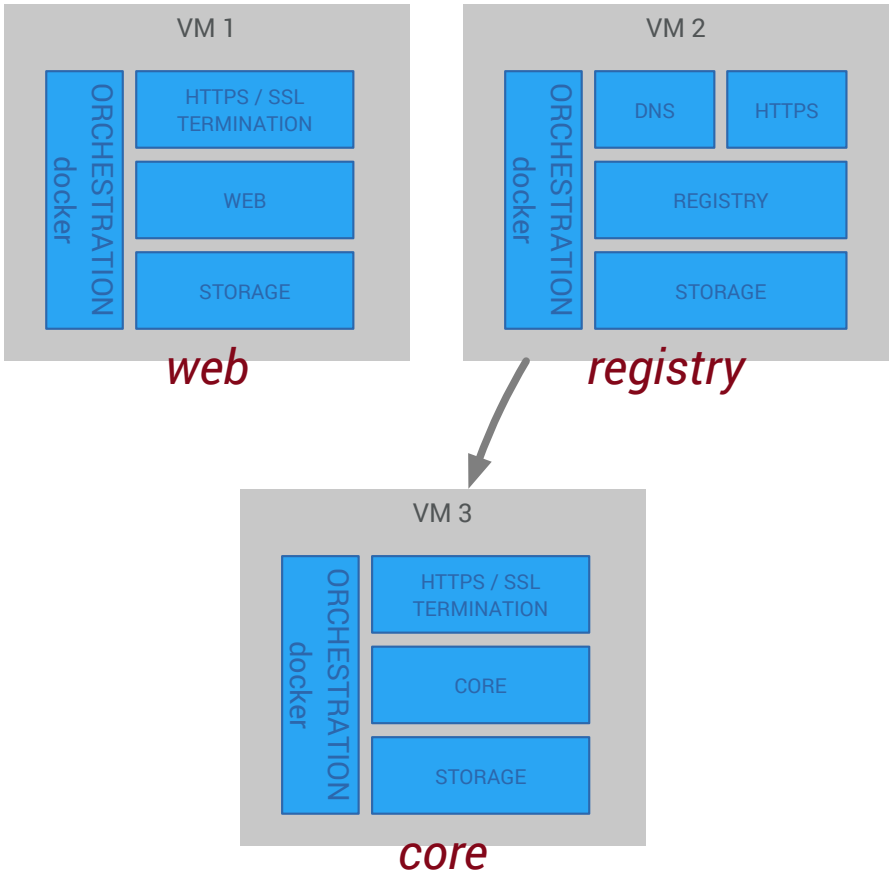
Pryv IO can be deployed in various ways, depending on requirements of your business case. This ranges from a development machine during a proof of concept phase where all components live on one virtual machine in a single location to a deployment spanning many machines across the globe. The present guide aims to allow the implementor to make a decision and deploy the Pryv that he needs. This section gives examples for the various scenarios and illustrates that choice graphically.

Development Installation



The diagram above shows deployment of Pryv IO on a single node. This can be ideal for development purposes or for testing. We don't recommend this for actual production use, but it can be the simplest way to get going.

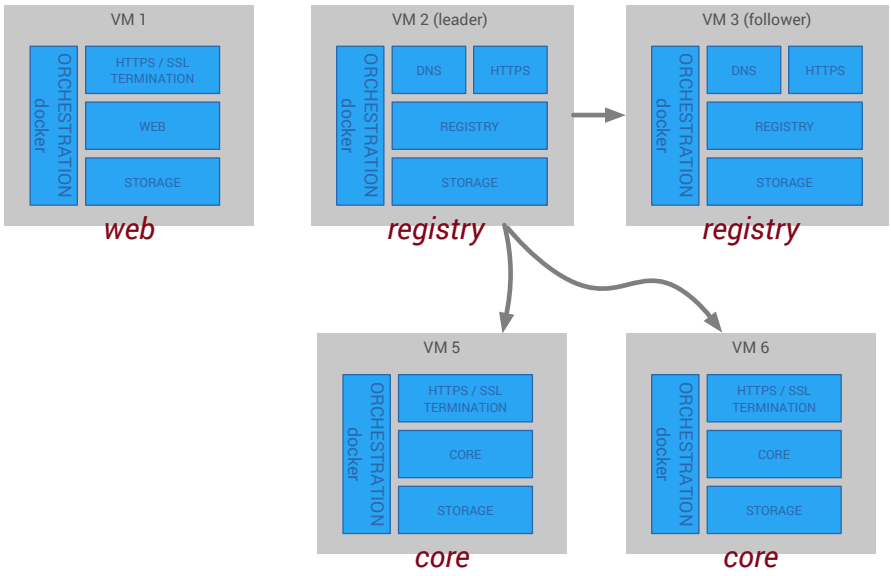
Proof of Concept Installation



Here we install all roles on separate machines. This can be an interesting deployment variant for creating proof of concept implementations: essentially, the capacities of the single core in this diagram are what they would be in a deployment with more machines; this allows testing and demonstrating Pryv realistically.

As we've mentioned above: The *web* role can be replaced by any static web-server like Cloudflare or Github. This is - where admissible - a useful variant that reduces administrative efforts.

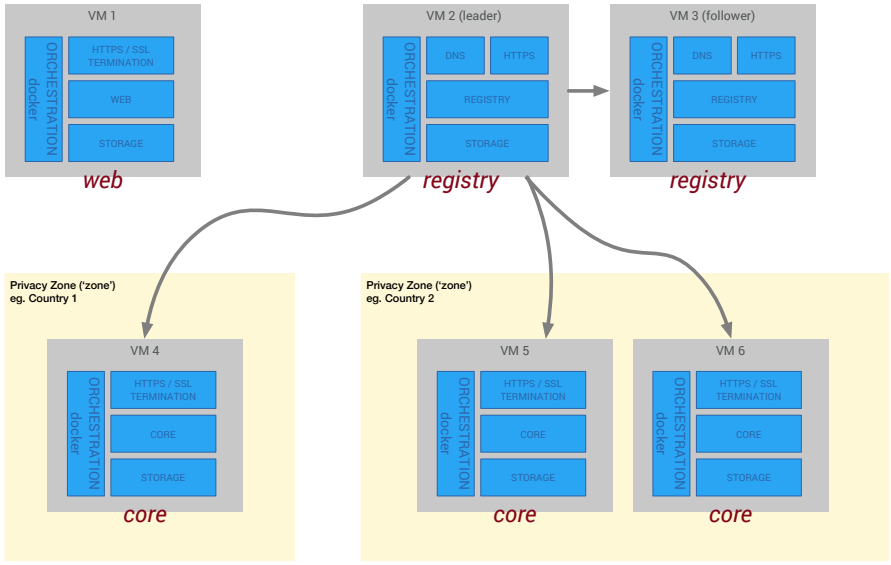
Partitioning for Load



When partitioning for load, multiple servers are in the *core* role and will receive user accounts in round-robin fashion. Any number of users can coexist on a *core*, up to the extreme of 1 user per machine. Please refer to section 'Deployment Design' for how to compute the amount of cores you will need for your particular load.

When partitioning for load, we recommend the creation of one or more follower nodes for the *registry* node. This avoids creating a single point of failure.

Partitioning for Privacy



The diagram above shows a Pryv IO system designed to partition data into multiple privacy zones. In practice, these will often correspond to countries (legislations) or smaller entities that handle data (and data privacy) differently.

Being able to store data in different locations might even be the reason you're using Pryv IO. In systems where Pryv IO coexists with other server components it is important to apply the same logic to all the components - e.g. an email server through which sensitive data transits would have to be deployed in multiple versions across privacy zones as well. Pryv offers consulting on the various legal and technical aspects of preserving user privacy and protecting data.

Keep in mind that the granularity of distribution in this kind of scenario is always the user account. In extreme cases a privacy zone might contain data for a single user only.

Business Requirements

The size and shape of any deployment will be determined by the business requirements that the Pryv IO cluster needs to meet. In this section, we aim to show what factors are relevant for designing a Pryv cluster.

Granularity

Pryv IOs fundamental entity is the user; data is kept vertically and not spread out. For this reason, the guidelines in this section will ask for requirements to be specified per user.

Data Production ('WRITE')

Metric	Your Values Here
Expected Write Requests Per Second (max rqps)	
Attachment Writes (max MB/s)	
Volume (data points per day)	
Volume (MB per day)	
Retention of data (years)	

The above table sums up the factors that influence the expected write load per user for your cluster. The first two metrics will influence the number of users that can be cohosted on a single core; the last two metrics will give you an estimation of disk space consumed per day per user.

Data Consumption ('READ')

Metric	Your Values Here
Expected Read Requests Per Second (max rqps)	
Number of Points retrieved per Request (scalar)	
Attachment Reads (rqps)	
Image Previews (rqps)	
Volume (data points per day)	
Volume (MB per day)	

This table should help you to quantify the load generated by reading data back per user.

Deployment Design

This section aims to guide you through the design of a Pryv IO deployment, using the key metrics you compiled in the last section. It should allow you to design a Pryv IO system on a macro-level. The next section ('System Requirements') will allow you to specify the sizing of the individual machine.

Single Node or Cluster?

Single Node Pryv IO deployments are primarily intended for development purposes or for simple proof of concept projects. Having only a single machine to maintain can be an advantage. On the downside a single node deployment cannot grow as easily as a 3-node cluster.

If any of the following apply, you should not create a single node cluster:

- Data stored is going to be production data that needs to persist.
- The integration made with other systems is part of the evaluation. Since links between the systems will look differently in a 3 node cluster, having a single node subtracts from the realism of the proof of concept.

If trying to decide for or against a single node deploy, please also check the formulas below for the amount of cores you might need, connected to the metrics in the previous section. If - even for testing - your target load exceeds a single core, as a consequence you will need a 3-node deployment at a minimum.

Cluster Sizing Considerations

Once a system gets bigger than a single node (see above), at least 3 machines will be required: one for the *web* role, one for the *registry* role and finally a server in the *core* role.

If your system is distributed among multiple privacy zones, you will need at least one core per such zone. Inside of every privacy zone, the number of cores should be derived from the following maximum values for a single core:

Metric	Max Performance of a Single Core
Write Requests Per Second	2000 requests per second
Users	< 10'000
Attachment Writes	Depends heavily on network path roughly speed of underlying storage system / 2
Data Points Per Day	Sustained write increases total data points per user, which will use more RAM and disk space.
Volume (MB per day)	See above.
Expected Read Requests Per Second	2000 requests per second Latency has a long tail distribution, depending on your query.
Number of Points retrieved per Request	Big (>1000 points) result sets should use cursoring. See Read Requests per Second.
Attachment Reads	600 requests per second
Image Previews	Depends on original image size and variations in target image size. 100 requests per second.

If your requirements for a single user exceed these quantities, you will need our high frequency module which is currently in a beta phase. Contact your sales representative if this is of interest to you.

Additionally, you should consider load distribution across your user base. Depending on homogeneity, you might add safety margins to the above numbers to allow for inter-user differences.

Users will be assigned to the core that has the least amount of users in any privacy zone. This results in a round-robin behaviour for a stable set of servers. In the presence of user deletion or when adding servers to an existing cluster, this will skew the distribution of users towards machines that have less users than the others.

System Requirements

The previous section should have allowed you to calculate how many cores to deploy in each privacy zone. The purpose of this section is to give you specifications for each machine in the three roles.

General requirements

The Pryv IO 1.3 release uses nginx to terminate inbound HTTPS connections. You should have obtained a wildcard certificate for your domain to that effect. Note that we will never ask you for your private key; you combine your certificate with our configuration upon install.

Please follow this link to find instructions on how to convert a certificate for nginx.

Single Node Deploy

The machine you use for your single node deployment should have the following (minimal) key specs:

Aspect	Minimal Requirement
RAM	4GB
CPU Cores	2
OS Disk	15GB
Data Disk	Depending on storage needs (*1)
Base OS	Linux / Ubuntu 16.04
Docker	>=1.13.0
Docker-Compose	Latest as of Nov 17
Service ports	tcp/443, udp/53
Administrative ports	tcp/22

*1: Since this is a development deploy, you might store only very little data.

Web Role Sizing

In a multinode deploy, here are the sizing specifications for a node in the 'web' role.

Aspect	Minimal Requirement
RAM	1GB
CPU Cores	1
OS Disk	15GB
Data Disk	not needed
Base OS	Linux / Ubuntu 16.04
Docker	>=1.13.0
Docker-Compose	Latest as of Nov 17
Service ports	tcp/443
Administrative ports	tcp/22

Registry Role Sizing

In a multinode deploy, here are the sizing specifications for a node in the 'registry' role.

Aspect	Minimal Requirement
RAM	2GB
CPU Cores	2
OS Disk	15GB
Data Disk	15GB
Base OS	Linux / Ubuntu 16.04
Docker	>=1.13.0
Docker-Compose	Latest as of Nov 17
Service ports	tcp/443, udp/53
Administrative ports	tcp/22

Disk space and RAM used on this node are proportional to the number of users registered in your Pryv IO instance. If you foresee a big number of user accounts (> 100'000), please get in touch with your Pryv sales support for numbers adapted to your needs.

Core Role Sizing

In a multinode deploy, here are the (minimal) sizing specifications for a node in the 'core' role.

Aspect	Minimal Requirement
RAM	4GB
CPU Cores	2
OS Disk	15GB
Data Disk	15GB
Base OS	Linux / Ubuntu 16.04
Docker	>=1.13.0
Docker-Compose	Latest as of Nov 17
VM Disk Encryption	Needed for better privacy
Service ports	tcp/443
Administrative ports	tcp/22

Here's a matrix that shows how various load situations affect the resource needs of your *core* servers:

Load Situation	Resource Needs
Many users on a core	Data Disk Space: Increase per user predictions. Add more machines beyond 10'000 users per core.
High Requests Per Second	CPU Cores: Increase to at least 4.
Image uploads and Previewing	CPU Cores: Increase to at least 4. RAM: Increase depending on needs.

Operational Concerns

This section will introduce additional operational concerns not covered by your Pryv base installation. We recommend implementing measures to address these topics in order to guarantee safe operation and traceability of issues.

System Hardening

We recommend you follow a system hardening guide for the operating system of your choice. This should include installing firewalls, denying SSH access using passwords and other measures that form best practices.

Administrators accessing a regulated system must themselves conform to the regulations and have received adequate training.

Backups

Pryv can be backed up using hot copies of the data disk where it is installed. Making a copy of private user data is regulated by law. Please make sure you know the ramifications of making backup copies.

High Availability (HA)

To render a core node highly available, we suggest using available products on the market to create a leader/follower-configuration with a hot standby. This can either be done by using Pacemaker/Heartbeat or redundant VMs with live monitoring.

At a basic level, the disk image of a Pryv IO core server can be hot-copied at any time and forms a consistent image; given an up-to-date version of such a disk image, a hot standby server can be launched.

Node Monitoring

Make sure you monitor key performance metrics of your Pryv nodes and keep a history of these metrics for later viewing. This helps in tracking down performance issues and is considered a best practice. Your metrics should include:

- Load, CPU (system, user, iowait, idle, load1, load5, load15)
- Disk (space left on devices, iops read and write)
- RAM (swapping activity, reserved, free)
- Network Interfaces (Packets, Bytes, Errors)

Outgoing Email

Pryv IO 1.3 requires an external service to send emails. This can either be done via Mailchimp or via the example service provided here: https://github.com/pryv/service-mail.

Support

If you need additional support in designing your Pryv IO deployment, please contact your sales contact or

sales@pryv.com.

Happy Installing Pryv IO! Remember, we are there to help. Sincerely,

Kaspar Schiess, Pryv; October 2018.