
# Europeana DSI 2– Access to Digital Resources of European Heritage

## MILESTONE

### MS7.1: Infrastructure Layer Version 1

| | |
|---|---|
| **Revision** | 1.0 |
| **Date of submission** | 28.04.2017 |
| **Author(s)** | Krystian Adamski, Tarek Alkhaeir, Marcin Heliński, Aleksandra Nowak, Marcin Werla, Paweł Woźniak (PSNC) |
| **Dissemination Level** | Public |

Co-financed by the European Union
Connecting Europe Facility

# REVISION HISTORY AND STATEMENT OF ORIGINALITY

## Revision History

| Revision No. | Date | Author | Organisation | Description |
|---|---|---|---|---|
| 1 | 10.04.2017 | Krystian Adamski, Tarek Alkhaeir, Marcin Heliński, Aleksandra Nowak, Marcin Werla, Paweł Woźniak | PSNC | Initial Version |
| 2 | 28.04.2017 | Krystian Adamski, Tarek Alkhaeir, Marcin Heliński, Aleksandra Nowak, Marcin Werla, Paweł Woźniak | PSNC | Final Version |

## Statement of originality:

This milestone contains original unpublished work except where clearly indicated otherwise.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.

The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained therein.

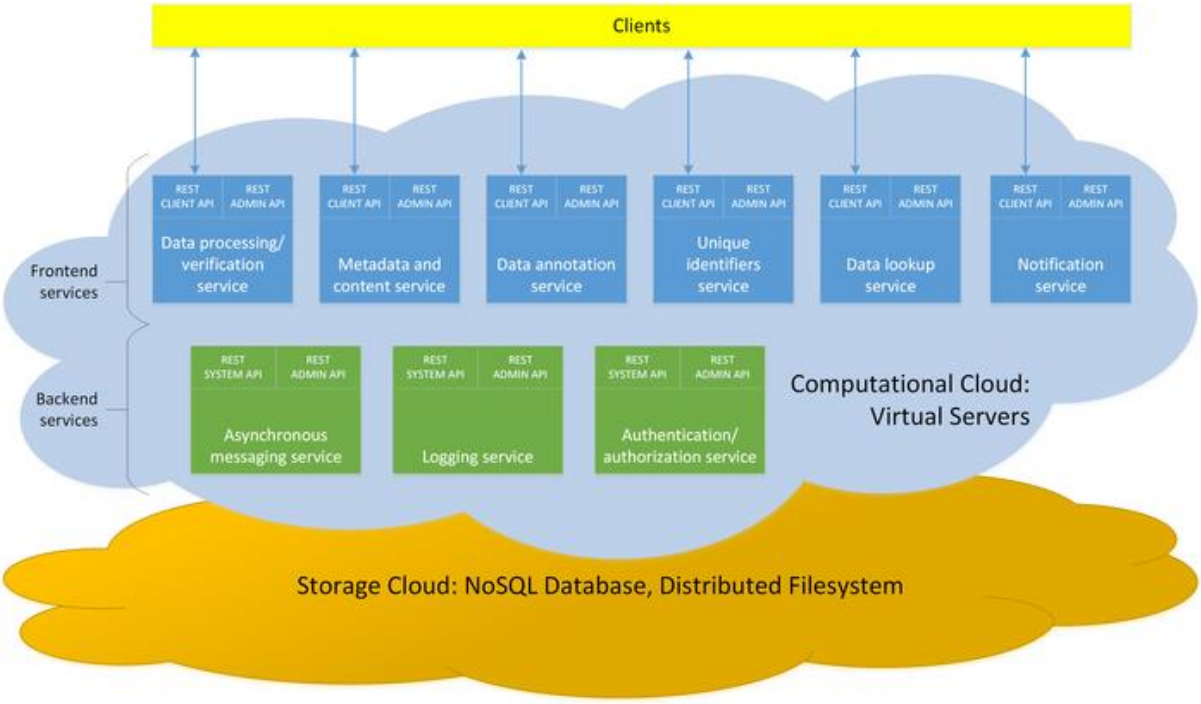
# Table of contents

# Introduction

This document describes the current state of the infrastructural part of Europeana Shared Services, initially created during Europeana Cloud project and further developed during Europeana DSI-1 and Europeana DSI-2 projects. Further in this document, current logical and physical architecture of the services is described together with the functionality outline (with emphasis on new developments related to tags and revisions) and with information about monitoring mechanisms, services performance and availability.

# Logical architecture of infrastructure services

The architecture of the Europeana Cloud was designed following the process of gathering requirements which is described in detail in the previous Europeana Cloud deliverable D2.2. Figure 1 from that deliverable, shown here for convenience, shows the main parts of this architecture.




SaaS) system and can be used similarly to other SaaS services available on the Internet today.

**Figure 1: Europeana Cloud Service Architecture**

To a client, the entire system appears as a Software-as-a-Service.

A client can use several services, each one providing an API for its own functionality. Services are designed to be stateless in order to allow horizontal scalability. They also follow the standard REST service design approach.

On resources level, the system consists of two types of cloud:

- **Computational Cloud**
  to provide computing capacity for services executed by clients and other services
- **Storage Cloud**
  to provide storage capacity for services deployed in the Computational Cloud

The **Computational Cloud** hosts all the services which can be used by clients and therefore it is the one which is explicitly contacted by them (frontend services, see below). Also backend services are hosted there (e.g. logging service). The **Storage Cloud** operates behind the scenes on behalf of the clients because all the services use it.

The services in the **Computational Cloud** are arranged in two layers:

- **Frontend Services**
  which are directly available for the clients of the system. They are also called functional services (blue) because they cater specific functionality used by clients.

- **Backend Services**
  which are internal and are not available directly for end users, but are used by other services for administrative purposes. They are also called system services (green).

Frontend Services The functionality of the Frontend Services can be divided into several groups of responsibilities:

- **Managing records identifiers** - to provide globally unique identifiers for cultural data records from diverse sources
- **Managing of metadata and content records** - To provide storage and access capabilities for cultural data records, consisting of data and metadata streams in many formats and versions To provide annotation capabilities for cultural data
- **Processing of the data** - to provide flexible, scalable and customizable cultural data processing capabilities
- **Presentation of the data** - to provide easy and reach access to high resolution images stored in Europeana Cloud.

## Access to the data

Files uploaded to Europeana Cloud Storage can be accessed using REST API of Metadata and Content Service. Huge part of the cultural heritage data are high resolution images. To fully use their potential and provide better user experience an additional service was developed - Europeana Cloud Image Service. The service provides access to images stored in Europeana Cloud. It is a compatible with International Image Interoperability Framework which provides a standardised method of describing and delivering images over the web. It supports images in JPEG2000 format. Users can display images from the service using compatible viewers on their websites or use an embedded viewer. Both options give high user experience (e.g. scroll to show details, manipulate fragments of the image, compare images or their fragments, etc.)

### Processing of the data

Data Processing Service is a part of Europeana Cloud dedicated to provide data processing facility. It consists of a set of build-in plugins providing different ways of data processing. Plugins available right now are useful to perform XML transformations (XSLT) and image transformations. The later plugin might be used to transform images to format compatible with Europeana Cloud Image Service described above.

# Main functional improvements of the infrastructure services: revisions and tags support[1]

One of the major clients for Europeana Cloud is Metis, a distributed multi-tier services framework for ingestion and delivery of content to Europeana. Prior to commencing the development of Metis, Europeana analysed the solution proposed by Europeana Cloud with regards to its functional and technical usability and identified various aspects of the system that needed to be improved both in terms of functionality. These changes affect the way that the content is represented within the Europeana Cloud system.

Europeana Cloud data model was designed in the beginning of the original project and can be found on the documentation pages[2]. While the model is descriptive enough for the use of storing and retrieving the content within Europeana Cloud, certain suggestions to improve efficiency have been made by the Metis development team. In short, in its previous iteration, the model exposed all the necessary hierarchies and relations required for the successful management of the stored records. However it imposed an overhead on the client applications that want to utilize Europeana Cloud, as extensive housekeeping operations needed to be performed on the Europeana Cloud clients in order to keep track of modifications in records and datasets, making the integration of the system complicated and reducing its efficiency. To ameliorate this issue stemming from the data modelling selected in Europeana Cloud the following specification was established collaboratively between the Metis team and the Europeana Cloud team. The latter approach removes the overhead of housekeeping data from the client side, making it efficient enough to build a system that can cater for both the needs of ingestion and presentation from third party services.

The new approach introduces a new level in Europeana Cloud record structure hierarchy - revisions and tags:

- A representation version may have one or more revisions
- A revision has an unique id, an owner, a creation timestamp and, optionally, tags
- Versions of records with the same revision have been processed by the same task execution in METIS workflow
- Tags - an additional information stored within revision (PUBLISHED, ACCEPTANCE, DELETED)

[1] Developed on the basis of Metis Technical Design Plan

[2] https://confluence.man.poznan.pl/community/display/ECLOUD/Europeana+Cloud+User+Documentation#EuropeanaCloudUserDocumentation-DataModel

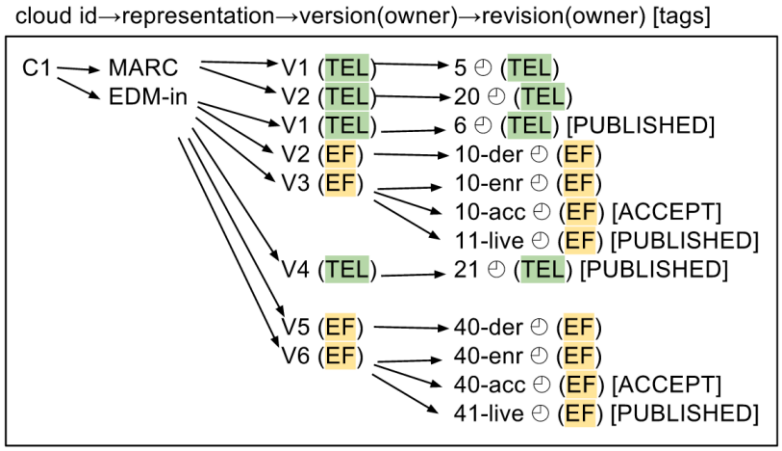



**Figure 2: record with two representations: MARC and EDM-in. Both of them have several versions. With every version there is a revision associated**

# Physical architecture of the infrastructure services

The installation of Europeana Cloud at PSNC consists of several groups of servers acting as an integrated system, called clusters, on which services of the system or 3rd party software are installed. Efficient clustering is critical for building a successful distributed computing system. On the one hand, clusters are able to execute resource-intensive tasks, such as a heavy query or a batch update, quickly because all cluster's machines work on the task altogether. On the other hand, the right balance between clusters should be created. Because resources are not unlimited, systems consisting of several clusters are to be built to avoid bottlenecks between clusters and skewed distribution of resources.

There are two types of machines used for the clusters:

1. Physical machines which provide the best of performance but do not allow for virtualisation (designated by rectangles)
2. Virtual machines with networked storage (designated by ellipses)

The infrastructure of Europeana Cloud consists of several independent components that act together to achieve high performance. We followed these principles to achieve a plausible result:

- The storage part of the system, the Storage Cloud, has to be performant as well as reliable. Therefore for it we used physical machines with a lot of processing power and significant number of hard drives.
    - The database cluster can be easily scaled out using the out-of-the-box ability of the system to scale when a new node is added.
    - The storage cluster cannot be scaled as easily because new nodes have to be configured. If scaling will be needed often, a standard scaling procedure will be prepared.
- The messaging system acts as intermediary between the services and will experience a lot of traffic. Thus it is designed an independent cluster built on physical machines. The messaging software has a built-in scaling mechanism so that this cluster can be enhanced when needed.

- The search index cluster consists of two machines, one of which is used for indexing new records and the other for searching. This in order to allow quick search operations while there is a massive indexing operation going on.
- Application services reside on their own cluster for quick identification of a performance bottleneck in one of them. Horizontal scaling mechanism should be designed to allow quick resolution of such bottlenecks, load balancing and failover. For the moment there is no natural clustering mechanism to group several physical services into one logical. It will be developed in the future and rely on Apache Zookeeper. This is on our list.

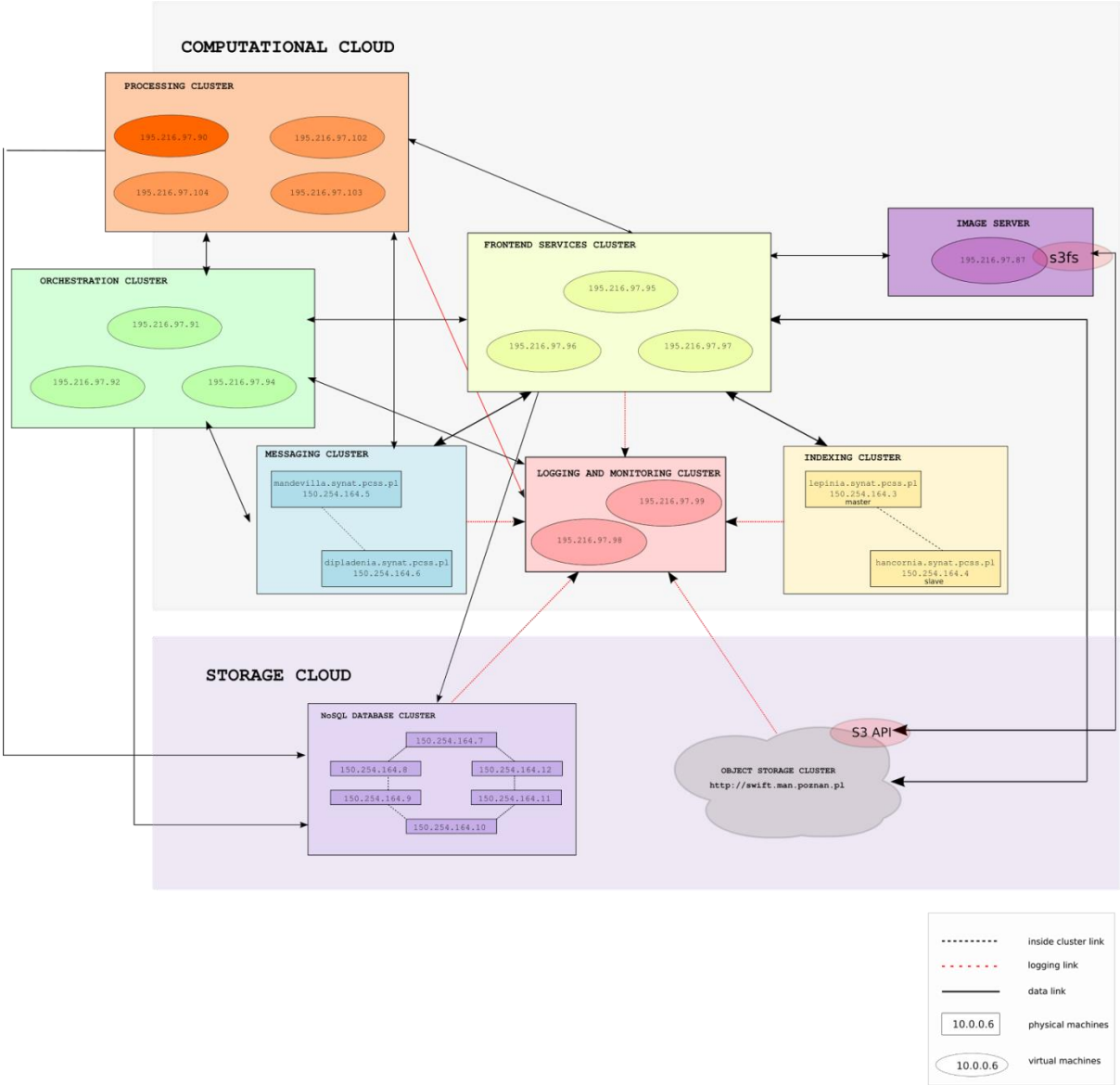



**Figure 3: Europeana Cloud Deployment Scheme**

# Services monitoring, performance and availability

To assure the highest availability of Europeana Cloud infrastructure and services are monitored by:
**Ganglia Monitoring System** - is a scalable distributed monitoring system for high-performance computing systems. The software is used to view live or recorded statistics covering metrics such as CPU load averages or network utilization for many nodes. One can also define and implement custom statistics. All the machines in the Europeana Cloud cluster are connected to the tool, that's why the team can easily monitor the health of the system and the usage of the resources. Ganglia reports are only available for the maintenance team.
The image below shows is an example report. It shows an aggregated load for all the application machines for the last hour.

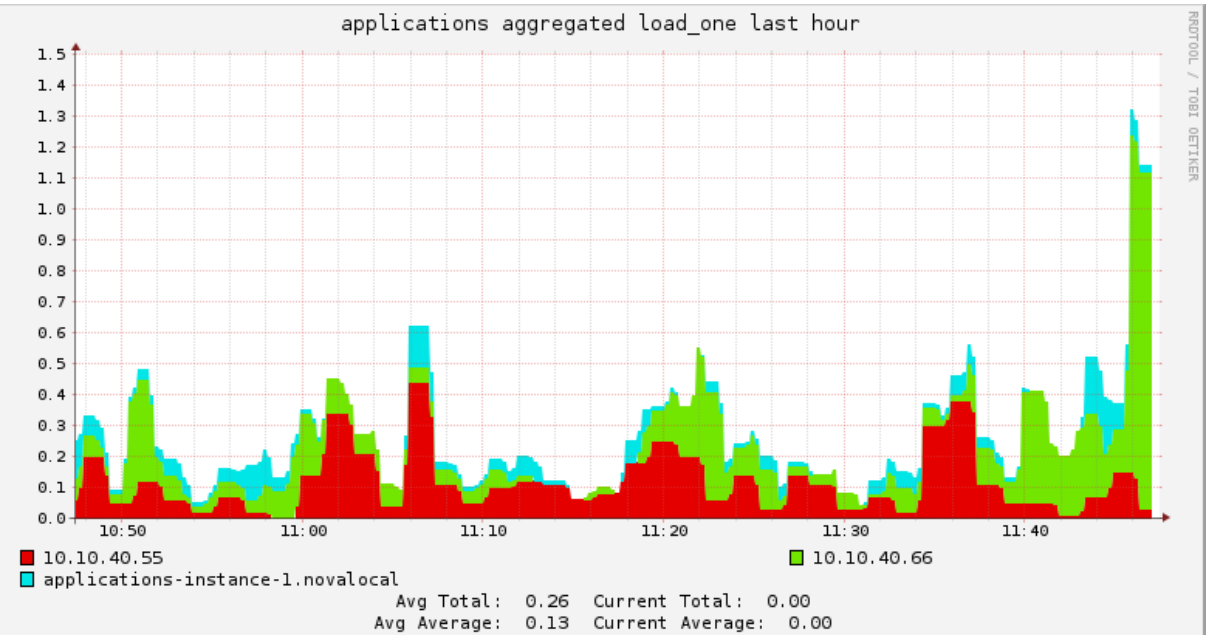



**Figure 4. An example of Ganglia statistics - aggregated load for application machines**

**Nagios -** is a tool that periodically runs health checks defined for every machine and service. In case the check failed a notification is send to support team, so they can take verify it and take the action needed to fix the problem. Based on the results of the checks availability statistics can be prepared. Nagios status and notifications are also available for the maintenance team only.
Figure below shows availability report for application machines for the period of last month.

Figures below present availability of the frontend services since the beginning of the year:

## Host 'prod_applications_instance_1'

01-01-2017 00:00:00 to 27-04-2017 14:42:08
Duration: 116d 13h 42m 8s

### Host State Breakdowns:

| State | Type / Reason | Time | % Total Time | % Known Time |
|---|---|---|---|---|
| UP | Unscheduled | 116d 12h 55m 29s | 99.972% | 99.972% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 116d 12h 55m 29s | 99.972% | 99.972% |
| DOWN | Unscheduled | 0d 0h 46m 39s | 0.028% | 0.028% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 0h 46m 39s | 0.028% | 0.028% |
| UNREACHABLE | Unscheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 0h 0m 0s | 0.000% | 0.000% |
| Undetermined | Nagios Not Running | 0d 0h 0m 0s | 0.000% | |
| | Insufficient Data | 0d 0h 0m 0s | 0.000% | |
| | Total | 0d 0h 0m 0s | 0.000% | |
| All | Total | 116d 13h 42m 8s | 100.000% | 100.000% |

**Figure 5: availability report for applications machine 1**

## Host 'prod_applications_instance_2'

01-01-2017 00:00:00 to 27-04-2017 14:36:02
Duration: 116d 13h 36m 2s

### Host State Breakdowns:

| State | Type / Reason | Time | % Total Time | % Known Time |
|---|---|---|---|---|
| UP | Unscheduled | 116d 12h 36m 8s | 99.964% | 99.964% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 116d 12h 36m 8s | 99.964% | 99.964% |
| DOWN | Unscheduled | 0d 0h 59m 54s | 0.036% | 0.036% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 0h 59m 54s | 0.036% | 0.036% |
| UNREACHABLE | Unscheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 0h 0m 0s | 0.000% | 0.000% |
| Undetermined | Nagios Not Running | 0d 0h 0m 0s | 0.000% | |
| | Insufficient Data | 0d 0h 0m 0s | 0.000% | |
| | Total | 0d 0h 0m 0s | 0.000% | |
| All | Total | 116d 13h 36m 2s | 100.000% | 100.000% |

**Figure 6: availability report for applications machine 2**

## Host 'prod_applications_instance_3'

01-01-2017 00:00:00 to 27-04-2017 14:29:34
Duration: 116d 13h 29m 34s

### Host State Breakdowns:

| State | Type / Reason | Time | % Total Time | % Known Time |
|---|---|---|---|---|
| UP | Unscheduled | 116d 12h 41m 40s | 99.971% | 99.971% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 116d 12h 41m 40s | 99.971% | 99.971% |
| DOWN | Unscheduled | 0d 0h 47m 54s | 0.029% | 0.029% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 0h 47m 54s | 0.029% | 0.029% |
| UNREACHABLE | Unscheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 0h 0m 0s | 0.000% | 0.000% |
| Undetermined | Nagios Not Running | 0d 0h 0m 0s | 0.000% | |
| | Insufficient Data | 0d 0h 0m 0s | 0.000% | |
| | Total | 0d 0h 0m 0s | 0.000% | |
| All | Total | 116d 13h 29m 34s | 100.000% | 100.000% |

**Figure 7: availability report for applications machine 3**

## Host 'prod_image_server'

01-01-2017 00:00:00 to 27-04-2017 15:26:14
Duration: 116d 14h 26m 14s

### Host State Breakdowns:

| State | Type / Reason | Time | % Total Time | % Known Time |
|---|---|---|---|---|
| UP | Unscheduled | 116d 13h 13m 54s | 99.957% | 99.957% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 116d 13h 13m 54s | 99.957% | 99.957% |
| DOWN | Unscheduled | 0d 1h 12m 20s | 0.043% | 0.043% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 1h 12m 20s | 0.043% | 0.043% |
| UNREACHABLE | Unscheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Scheduled | 0d 0h 0m 0s | 0.000% | 0.000% |
| | Total | 0d 0h 0m 0s | 0.000% | 0.000% |
| Undetermined | Nagios Not Running | 0d 0h 0m 0s | 0.000% | |
| | Insufficient Data | 0d 0h 0m 0s | 0.000% | |
| | Total | 0d 0h 0m 0s | 0.000% | |
| All | Total | 116d 14h 26m 14s | 100.000% | 100.000% |

**Figure 8: availability report for Image Service machine**

**Pingdom -** can also perform defined health checks and send notifications. Health checks are defined in Europeana Foundation's Pingdom instance, which can be accessed publicly. Right now Pingdom only reports about the health of the Europeana Cloud Image Service. Integration with the rest of services is planned.

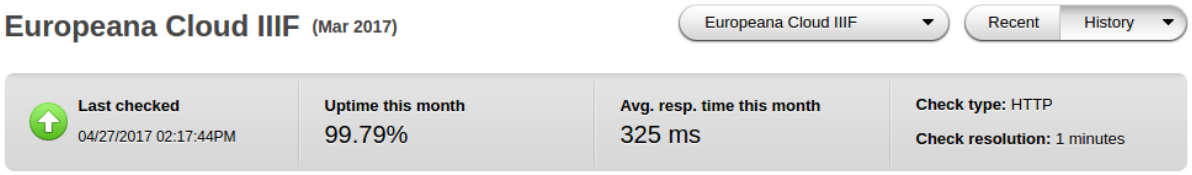



**Figure 9: Pingdom uptime status for Image Service for March 2017**

**Logstash/Banana -** is a powerful and configurable set of tools that are able to extract statistics of the services performance using logs reported by the software. It is possible to detect number and type of received requests, number of errors, client based statistics etc. Figure X presents an example board for Europeana Cloud application machines.

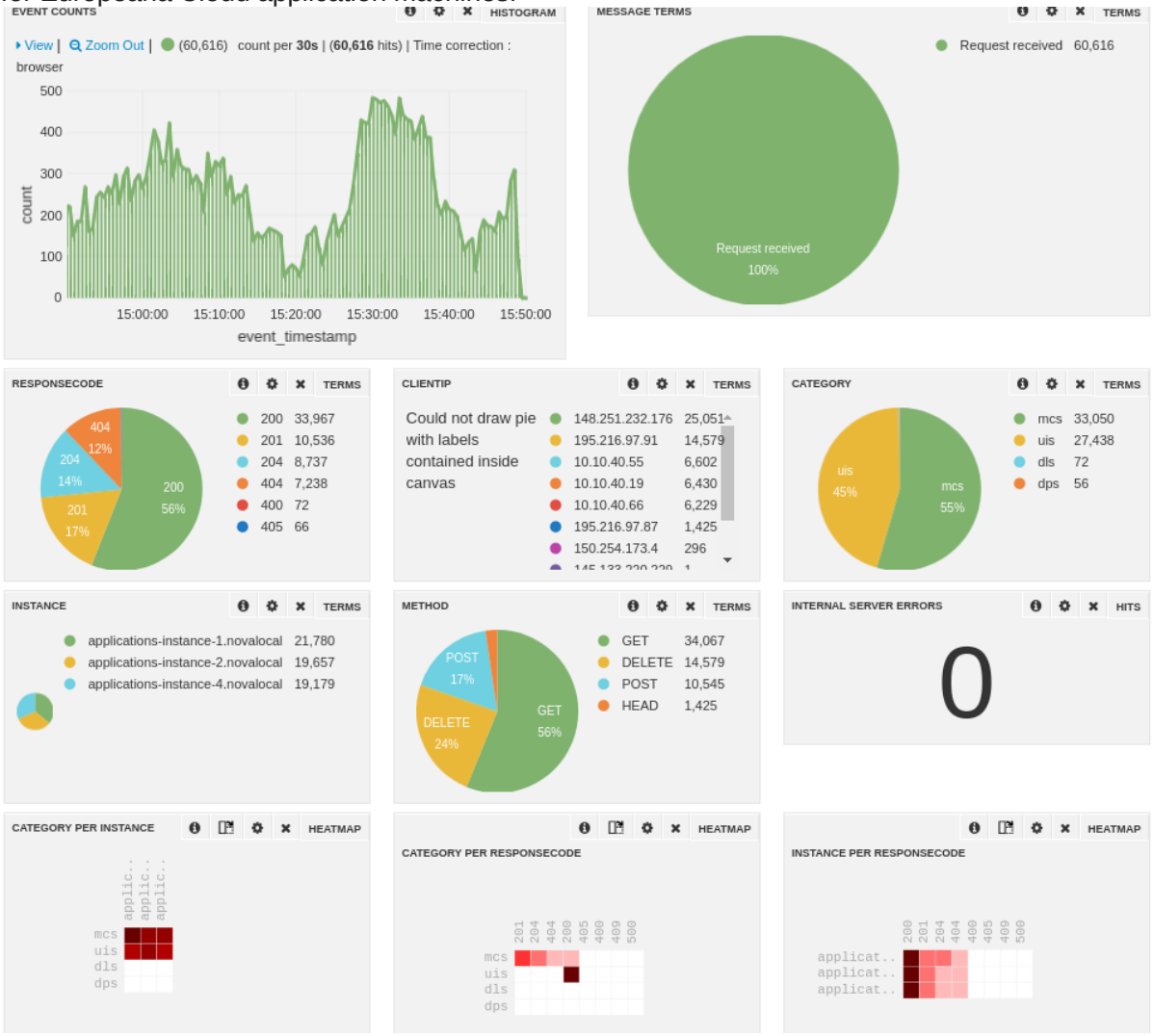



**Figure 10: Banana board for Europeana Cloud**