BalticLSC Platform User Requirements Specification

Detailed list of requirements for the BalticLSC Platform

Version 1.0

Priority 1: Innovation

Warsaw University of Technology, Poland
RISE Research Institutes of Sweden AB, Sweden
Institute of Mathematics and Computer Science, University of Latvia, Latvia
EurA AG, Germany
Municipality of Vejle, Denmark
Lithuanian Innovation Center, Lithuania
Machine Technology Center Turku Ltd., Finland
Tartu Science Park Foundation, Estonia

BalticLSC Platform User Requirements Specification

Detailed list of requirements for the BalticLSC Platform

Work package	WP3
Task id	A3.3
Document number	O3.3
Document type	Requirement Specification
Title	BalticLSC Platform User Requirements Specification
Subtitle	Detailed list of requirements for the BalticLSC Platform
Author(s)	Magnus Nilsson-Mäki (RISE)
Reviewer(s)	Daniel Olsson (RISE), Agris Šostaks (IMCS)
Accepting	Michał Śmiałek (WUT)
Version	1.0
Status	**Final version**

Interreg Baltic Sea Region
EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND

History of changes

Date	Ver.	Author(s)	Change description
16.10.2019	0.01	Magnus Nilsson-Mäki	Initial structure.
21.11.2019	0.02	Magnus Nilsson-Mäki	Updated
12.12.2019	0.1	Magnus Nilsson-Mäki	Changed after review update.
17.12.2019	0.11	Agris Šostaks	Second review
04.02.2020	1.0	Magnus Nilsson-Mäki	Final approved version

Executive summary

This document describes the requirement on the platform from a platform administrator point of view.

Table of contents

1. Introduction

1.1 Objectives and scope

The objective of the document is to describe the requirements on the platform. It describes different use-cases and technical requirements regarding installation, user handling, security, backup etc. Its intended for a BalticLSC Supplier (Resource provider).

1.2 Relations to other documents

The content of this document is based on dialogue in meetings held by BalticLSC partners during Q1-Q2 2019 and is based on document O3.1 Baltic Environment Vision. Individual partner overviews are available at Baltic LSC document Dropbox depository: Dropbox\BalticLSC\WP3_User_requirements\A3.1 End-user requirements elicitation workshops.

1.3 Intended audience and usage guidelines

The User Requirements as described in this document (O3.2) is intended to be used as basis for future platform and software development, i.e as input to Baltic LSC Platform Technical Documentation. (Baltic LSC Output 4.3A) as well as for describing the requirements on the platform for local First Level Control and Baltic Sea Region Program.

2. System setup

This section specifies minimum requirements for the hardware of the BalticLSC platform component, and also some general suggestions on how to administer and monitor the platform after installation.

2.1 Installation

2.1.1 Hardware

Its not a requirement, more a recommendation, to always use latest firmware and BIOS in your servers, because when they are in production it can be difficult or cause production disturbances when you do an upgrade afterwards. Regarding the network, the faster internet connection the better of course. Internally in your cluster, use at least 10Gb.

Role	**Use at least**
Kubernetes worker node	3 physical servers with as many GPU's as possible
Kubernetes master node	3 servers (physical or virtual)
Rancher nodes	3 servers (physical or virtual)

2.1.2 Disk recommendations

Regular hdd's is sufficient for the OS, but for kubelet it is good to use ssd's, mount /var/lib/kubelet on an ssd of at least 1TB, and use LVM on the worker nodes. As OS disk 500GB hdd is enough on worker nodes. On Kubernets masters and Rancher nodes 100 GB is enough as system disk.

2.1.3 OS and software components

Every component should be installed in HA (High Availability) mode (Kubernetes, Rancher, 3rd party load balancer for incoming traffic, etc). Use installation guide/instructions provided by RISE. Then you get required kubernetes version etc. Use Ubuntu version 18.04 or higher as OS (preferred version provided by RISE). An obvious requirement since kubernetes is to be used, the underlying OS must have the Docker runtime environment installed.

2.2 Performance requirements

Kubernetes and Rancher are rather CPU intense, so the newer equipment the better, both Kubernetes and Rancher have their own minimum requirements on their websites, these recommendations are from own experience:

Role	**Recommended by RISE to use in BalticLSC**
Kubernetes worker node	Dual socket 6 core CPU, 128 GB RAM
Kubernetes master node	Dual core CPU, 16 GB RAM
Rancher nodes	Dual core CPU, 16 GB RAM

2.3 Backup and redundancy

In a Kubernetes HA setup the etcd database is automatically synced between master nodes, so no specific file or database backup is needed, same goes for Rancher nodes. However, it's a good idea to backup the entire machines if they are running in a virtual environment such as VmWare or similar. If

you are using ESXi (free version) there are simple backup scripts like ghettoVCB that you can download and use for backup of everything once a week or so.

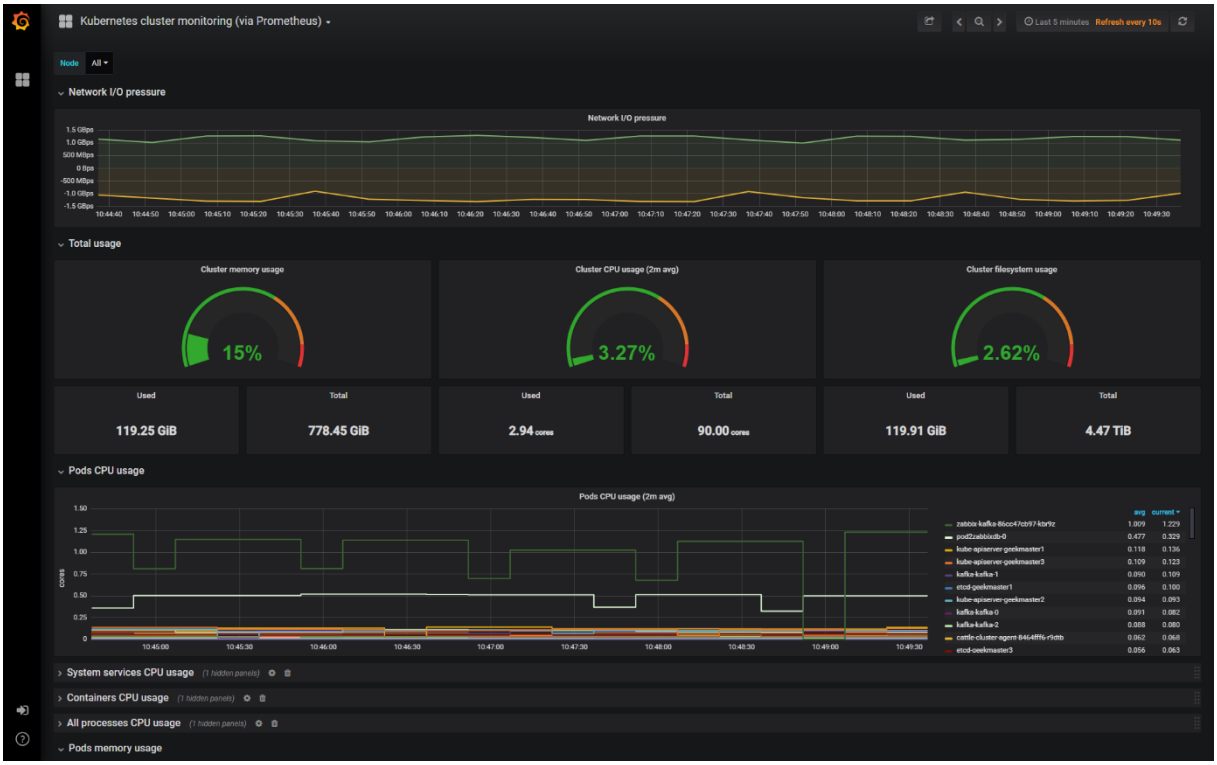
2.4 Monitoring of hardware and software

Rancher has built-in support for monitoring, install Prometheus from the catalogue and you get detailed monitoring for all levels of deployments, and hardware. See examples below:

Node monitor:

Pod monitor:

Cluster monitor:

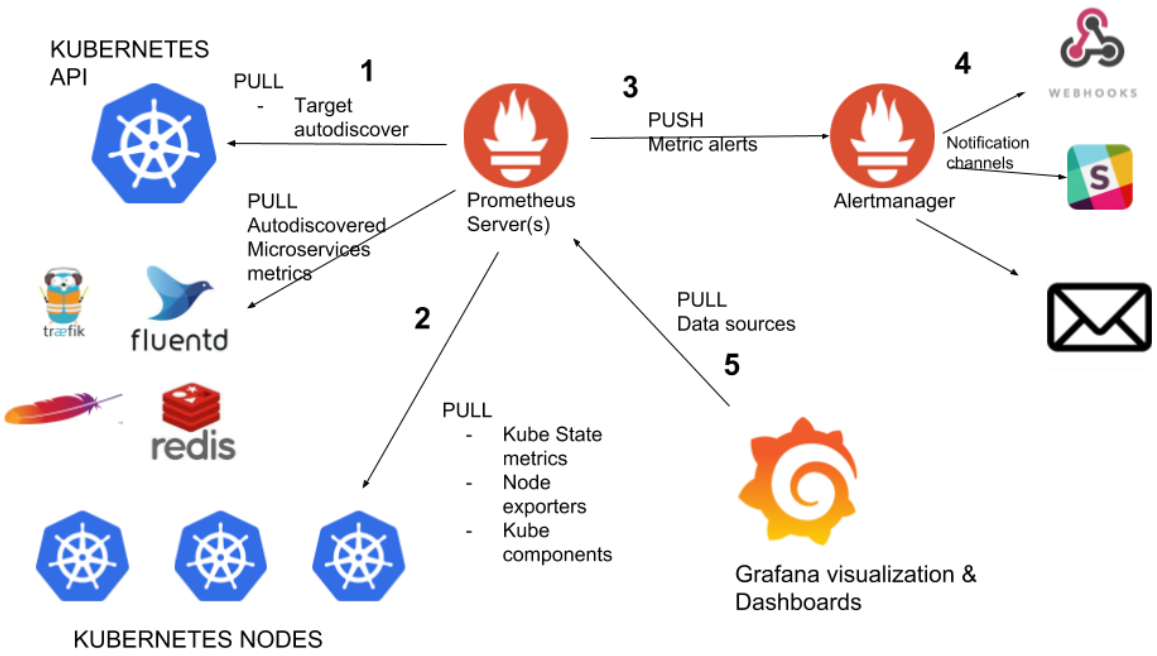


Prometheus also has an Alert Manager which can send alerts with numerous protocols.

3. System functions and features

3.1 Resource usage tracking

To enable business models, hardware resource usage tracking is mandatory. This means that the amount of resources (CPU, GPU, RAM, etc.) that a container consumes during its lifetime is tracked and stored (eg. Prometheus):



Explanation of the different steps:

1. Prometheus auto-discovers as much as it can.
2. Metrics related to Kubernetes and nodes are collected; CPU, memory, network, pods, etc.
3. Rules can be made to trigger alerts.
4. Alertmanager component configures receivers.
5. Grafana can pull metrics from any number of Prometheus servers and display panels and dashboards.

3.1.1 Resource logging

Logs from resources (containers) and nodes will only be available for the System Administrator of the cluster.

3.1.2 Resource error handling

In case of sudden errors, appropriate actions must be taken. Affected people of a specific resource must be immediately notified of current condition. Eg. a pod dies, software platform user must be informed, or if a node dies, the site System Administrator will get a mail or text message, or both, depending on severity.

3.1.3 Resource recovery

The crucial parts of the Kubernetes cluster (the master nodes), will run in a high-availability-configuration, meaning the etcd-database will be on 3 different physical hosts, and each master will have a backup of the entire node. A worker node will not be backuped. In case of worker node failure, all jobs on that particular node will fail. A worker node will not be backupd, it will have to be reinstalled.

3.2 User isolation and quota

Users of the platform must be isolated from each other, and all jobs for a user must be secured and sandboxed. This means that one user shall not be able to affect another user in any way, and same goes between the users jobs. The Platform must also support resource quotas and limits. This means that it must be possible to specify how much resources (see above) a user is allowed to allocate, as well as controlling that the user does not use more than requested. When this is specified, the platform can schedule the job on a suitable node. Kubernetes builtin scheduler takes care of this.

3.3 Security

As mentioned in 3.2 all users should be isolated from each other and not be able to read/write/execute in another project than your own, however, the administrator of the platform will have all permissions in all projects for all other users.

3.4 REST API

There must exist a way to programmatically use the compute resources of the Platform.A REST API must therefore be provided which will beused by the users of the Platform. The most obvious user is the BalticLSC Software.

3.5 Typical use-cases

Event	**Expected outcome on platform**
A new worker node added to the Kubernetes cluster	Node integrated and available for users. Statistics collection for node started (Prometheus integrated)
Request for worker node statistics	Statistics replied to user
A worker node cordoned	No new jobs scheduled to this node
Request for worker node details	Node details replied
New calculation job submitted from BalticLSC Software	Job is deployed on BalticLSC Platform and resource usage is tracked
Calculation job deleted from BalticLSC Software	Job deleted from Kubernetes and resources released and available again
Resource usage on a calculation job from BalticLSC Software requested.	Wanted data returned. E.g. CPU usage, memory usage, execution time.