



Polarion®

# Enterprise Setup Guide

**Multi-server Installation &  
Configuration**

## **Polarion Enterprise Setup Guide**

Version: 2016 SR1

Revision: 823154 - 20160615

Copyright © 2016 Polarion Software

Permission is granted to use, copy, and distribute this document provided that the content is not modified and the above copyright notice and this statement appear in all copies.

Polarion is a registered trademark of Polarion Software. Polarion ALM, Polarion REQUIREMENTS, Polarion QA and Polarion VARIANTS are trademarks or registered trademarks of Polarion Software.

### **Acknowledgements**

Apache and Subversion are trademarks of Apache Software Foundation.

Java is a trademark of Oracle Corporation.

Microsoft and Microsoft Windows are registered trademarks of Microsoft Corporation.

Linux is a trademark of Linus Torvalds.

All other trademarks referenced are property of their respective owners.

No relationship or endorsement of or by Polarion Software is implied by the appearance of third party trade or service marks in Polarion documentation.

# Polarion Enterprise Setup Guide

---

1	Terminology .....	3
2	Overview .....	3
3	Details .....	3
4	Requirements .....	4
4.1	Server Software Requirements .....	4
4.1.1	Requirements for Windows installation .....	4
4.2	Server Hardware Requirements .....	4
4.2.1	Coordinator Server .....	5
4.2.2	Stand-alone Instance Server .....	5
4.2.3	Cluster Instance Server .....	6
4.2.4	Shared Services Server .....	6
4.2.5	Example Hardware Configurations for Instance .....	7
4.3	License requirements .....	7
5	Installation Use Cases .....	7
5.1	Common terms .....	8
5.2	Setting up a Cluster from New Installations .....	8
5.2.1	Configuring the Cluster's Coordinator .....	9
5.2.1.1	License Deployment on Coordinator .....	10
5.2.2	Configuring the Cluster's Shared Services .....	10
5.2.3	Configuring the Cluster's Nodes .....	12
5.2.3.1	Synchronizing Time on Cluster Nodes .....	13
5.2.4	Configuring the Cluster's Activation Application .....	13
5.3	Setting Up With Multiple Stand-alone Instances .....	13
5.3.1	Configuring the Coordinator for Multiple Stand-alone Instances Setup .....	14
5.3.2	Configuring Instance1 .....	14
5.3.3	Configuring Instance2 .....	15
5.3.4	Access URLs for Multiple Stand-alone Instances .....	16
5.4	Migrating From a Pre-2014 "Multi-Instance" Installation .....	16
5.4.1	Configuring the Coordinator .....	16
5.4.2	Migrating a Remote Instance to a Non-clustered Stand-alone Instance .....	17
5.4.2.1	Checking the Migration .....	17
5.4.3	Moving Local Instances for the Multiple Stand-alone Instances Setup .....	17
5.5	Updating a Multiple Stand-alone Instances or Cluster setup .....	18
6	Configuring Shared Data .....	18
6.1	Linux Configuration .....	19
6.1.1	NFS Configuration .....	19
6.2	Windows Configuration .....	19
6.2.1	CIFS/Samba Share Configuration .....	20
6.2.1.1	Shared Services Machine .....	20
6.2.1.2	Node Machines .....	20
7	Security Options .....	20
7.1	Recommended Security Options .....	21
7.2	Advanced Security Options .....	22

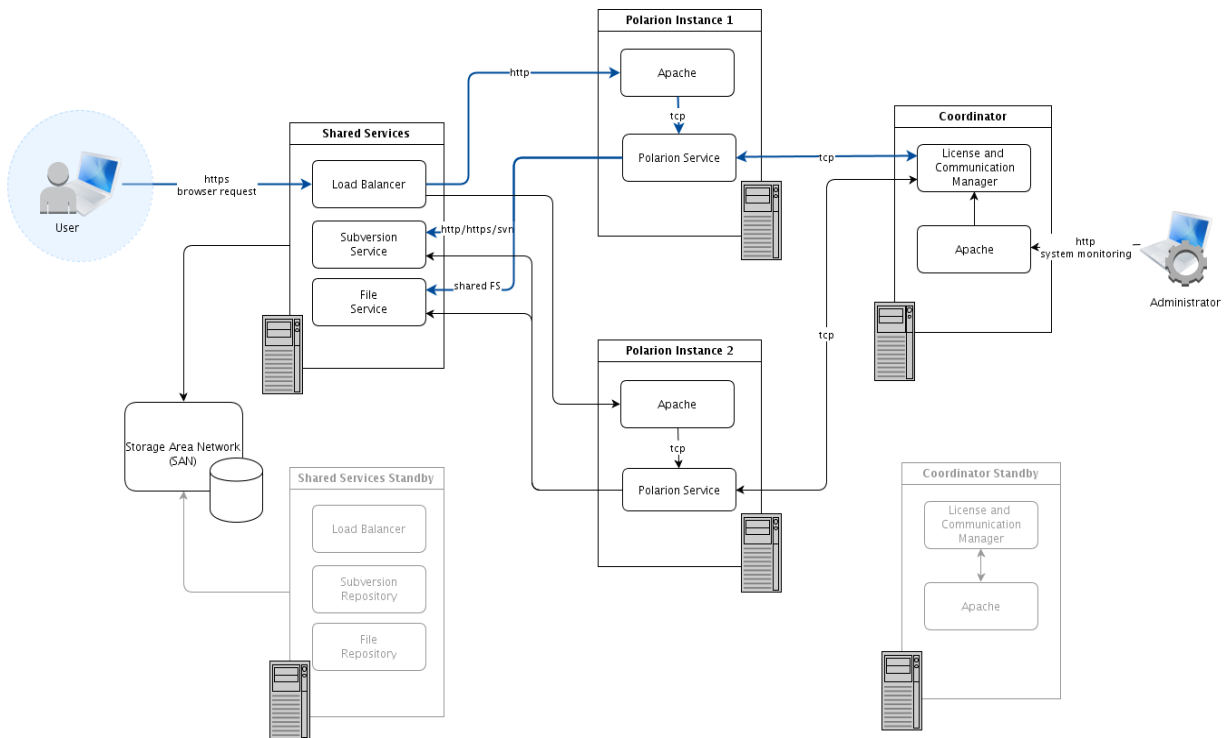
7.3 Authentication for Server Monitoring .....	23
8 Notes .....	23
9 Troubleshooting .....	24
10 Appendix: Polarion Instance Architecture .....	24

## 1 Terminology

- **Instance** - Machine with one running Polarion installation.
- **Stand-Alone Instance** - An Instance, with its own repository on the same machine (i.e. not clustered), exposed to the user as one Polarion Server.
- **Cluster** - A group of Instances accessing the Shared Services, exposed to the user as a one logical Polarion Server.
- **Coordinator** - A specially-configured Polarion installation that manages communication and licenses among Instances.
- **Shared Services** - A machine that hosts the Subversion repository, shared data and load balancer (user entry point). There is one Shared Services per Cluster.

## 2 Overview

The following figure shows one clustered setup and several stand-alone Instances, each with its own repository.



## 3 Details

A cluster setup requires one dedicated physical or virtual machine for each Instance. Coordinator and Shared Services must be dedicated machines as well.

## Coordinator

- Distributes tasks that need to be executed on exactly one Instance in the Cluster.
- Serves as Licensing Server for all Instances connected to the Coordinator.
- Provides a single entry point to all logical Polarion Servers that are connected to the same Coordinator.
- Reconfigures the Load Balancer if some of the Instances are offline.

## Shared Services

- Provides the Load Balancer that forwards users to a specific Instance in the Cluster.
- Entry point for one Cluster (on the picture above it is entry point for *Cluster*).
- Provides a File Repository shared by all Instances in the Cluster.
- Serves the Subversion repository, which contains the data for the clustered logical Polarion Server.

## Instance1, Instance2

- Machines running Polarion Service, connected in a Cluster and all configured to use the same Shared Services.
- Every instance in the Cluster has its own Polarion data (indexes, object maps), and PostgreSQL database. (Shared database is not currently supported.)

## 4 Requirements

Several virtual or physical machines are needed: one for Coordinator, one for every Instance (stand-alone or from Cluster) and one Shared Services per Cluster. The following sections briefly describe the requirements for all these servers.

### 4.1 Server Software Requirements

Same for all the machines as described in the Polarion Installation Guides (for Windows and Linux), with one exception: minimum Apache HTTP Server version is 2.2.17 (latest 2.4.x is recommended).

Although the Coordinator machine does not really need Subversion, it is still recommended to use the standard Polarion Installer to install Polarion on it. It will install Subversion on the Coordinator, and can just remain there.

Coordinator, Nodes, and Stand-alone Instances must all be running the same version of Polarion.

#### 4.1.1 Requirements for Windows installation

A Polarion clustered setup in a Microsoft Windows environment requires the following:

- MS Active directory.
- A DNS service is recommended, but you can also use static IP addresses (the same subnet is expected).
- Testing for proper network configuration using 'ping' between all hosts in a cluster.
- A domain user (example: *yourDomain\polarion*).
  - For shared Services: CIFS/Samba requires a domain User (example: *yourDomain\polarion*).
- A mail server or mail GW is required for sending e-mail notifications.
- For proper configuration you will need to open ports as described in section 4.2 below.

### 4.2 Server Hardware Requirements

Generally the requirements for all server machines are similar to those described in the Polarion Installation Guides for

**Linux** and **Windows**. All machines must be connected by a fast 1 Gbps low-latency (< 10 ms) intranet network.

#### 4.2.1 Coordinator Server

Requirement	Description
CPU	1 - 2
RAM	1GB
Disk Space	10GB
FQDN	e.g. <i>coordinator.mycompany.com</i>
Access from Clients (http(s))	<p>Coordinator provides signpost and server monitoring pages. Choose the port (usually 80 for http, 443 for https), configure Apache and Polarion to use it.</p> <p><b>Related Polarion properties:</b>  <i>base.url=http(s)://host:port</i> (FQDN or IP address)</p>
Access from Instances	<p>Communication between instances and Coordinator (ZooKeeper) takes place on TCP/IP port specified by property <i>com.polarion.zookeeper.port</i> of the Coordinator, by default it is 2181. This port on Coordinator host must be accessible by all instances.</p> <p><b>Related Polarion properties:</b>  <i>com.polarion.zookeeper.port=port#</i> (on Coordinator)  <i>com.polarion.zookeeper=host:port#</i> (on Instances)</p>

#### 4.2.2 Stand-alone Instance Server

Requirement	Description
CPU	4 - 32
RAM	4GB, recommended 8GB or more
Disk Space	20GB, recommended 250GB or more
FQDN	e.g. <i>myserver1.mycompany.com</i>
Access from Clients (http(s))	<p>Choose the port (usually 80 for http, 443 for https), configure Apache and Polarion to use it.</p> <p><b>Related Polarion properties:</b>  <i>base.url=http(s)://host:port</i> (Must be FQDN or IP address)</p>
Access to Subversion	<p>The same as in the case of a simple installation. There should be http(s) access from clients (end users), svn protocol access is recommended for fast local access by system user.</p> <p><b>Related Polarion properties:</b>  <i>repo=http(s)://host:port/repo</i>  <i>repoSystem=[svn/file/http(s)://host:port</i></p>

### 4.2.3 Cluster Instance Server

Requirement	Description
CPU	4 - 32
RAM	4GB, recommended 8GB or more
Disk Space	20GB, recommended 250GB or more
Time Synchronization	Well synchronized system time with all other cluster instances.
Access from Load Balancer	<p>Load balancer needs to be able to redirect the requests to the cluster instances using on http(s) port where the Polarion is running.</p> <p><b>Related Polarion properties:</b>  <i>base.url=http(s)://host:port</i> (Must be FQDN or IP address)  <i>com.polarion.loadBalancer.workerUrl=http://host</i></p>
Communication between Cluster Instances	<p>RPC communication between cluster instances takes place on TCP/IP port specified by property controlPort of the instances. All instances of the cluster have to be able to access control ports of all other instances in the cluster.</p> <p><b>Related Polarion properties:</b>  <i>controlPort=port#</i>  <i>controlHostname=host</i> (Must be FQDN or IP address)</p>

### 4.2.4 Shared Services Server

Requirement	Description
CPU	2 - 8
RAM	2GB, recommended 4GB or more
Disk Space	20GB, recommended 250GB or more
FQDN	e.g. <i>myserver2.mycompany.com</i>
Access from Clients to Load Balancer (http(s))	<p>Entry point to the cluster. Choose http(s) protocol, configure Apache, adjust configuration of cluster instances.</p> <p><b>Related Polarion properties:</b>  <i>base.url=http(s)://host:port</i> (on instances - Must be FQDN or IP address)</p>
Access from Coordinator to Load Balancer Manager (http(s))	<p>Coordinator communicates with load balancer manager via http(s). Configure load balancer manager application location in Apache.</p> <p><b>Related Polarion properties:</b>  <i>com.polarion.loadBalancer=http(s)://host/balancer-manager</i> (on cluster instances)  <i>com.polarion.cluster.#ClusterId#.loadBalancer.user=</i> (on coordinator)  <i>com.polarion.cluster.#ClusterId#.loadBalancer.password=</i> (on coordinator)</p>
Shared Folder	<p>(Linux paths are used, use analogical paths on Windows.) Folder <i>/opt/polarion</i> of Shared Services has to be mounted as <i>/opt/polarion/shared</i> on all cluster instances. <i>This folder sharing should be set up after installation of Polarion.</i> User "polarion" on all nodes must have read access for <i>/opt/polarion/shared/**</i>, and write access at least for:</p>



	<pre> /opt/polarion/shared/data/svn/* /opt/polarion/shared/data/BIR/** /opt/polarion/shared/data/RR/** /opt/polarion/shared/data/workspace/** (/ * means files inside directory, / ** means everything including subdirectories recursively)  Files created by user on behalf of which polarion service is running ("polarion") on any node in /opt/polarion/shared must be readable by user on behalf of which Apache server is running on the shared services.  <b>Related Polarion properties:</b> com.polarion.shared=/shared/directory/path (on instances) </pre>
Access to Subversion	<p>From clients (end users) and each instance of the cluster, the Subversion repository must be accessible. Either http(s) or svn protocol can be used (svn is recommended for fast access by the system user).</p> <p><b>Related Polarion properties:</b>  repo=http(s)://host:port/repo (on instances)  repoSystem=[svn/http(s)]://host:port (on instances)</p>

#### 4.2.5 Example Hardware Configurations for Instance

	S	M	L	XL	XXL
Operating System	32 or 64-bit	64-bit	64-bit	64-bit	64-bit
CPU Cores	4	8	8	16	32
GB RAM (dedicated to Polarion)	8 (4)	16 (8)	32 (16)	64 (32)	128 (64)
Storage for Polarion	250GB+	500GB+	1TB+ (SCSi or similar)	1TB+ (RAID 10, NAS, SAN)	1TB+ (RAID 10, NAS, SAN)
# Projects	< 100	> 100	> 300	500	500
# Users	< 50	< 150	> 150	> 300	> 500

Make sure that there is enough RAM available to the OS for file-caching. If SVN is hosted on a different machine, more memory could be allocated for the Polarion process.

#### 4.3 License requirements

If you will host Polarion on your company's infrastructure, you must provide all physical hardware and/or virtual machines needed for the setup you want to implement (see Chapter 5 *Installation Use Cases*), and obtain a license for the instances you will run. If you utilize Polarion's cloud-based hosting services, you need to order a virtual server for each Instance of a clustered or multiple stand-alone instance configuration.

Every node in a cluster or server in a Multiple Stand-alone Instances setup counts towards the "multiple instances" limit set in the license. Please [contact Polarion Software](#) for assistance with any licensing questions.

## 5 Installation Use Cases

This chapter describes simple use cases: a *Cluster* and a *Multiple Stand-alone Instances* setup. A subsequent chapter will describe how to migrate a pre-2014 "multi-instance" setup to the new configuration (released with version 2014). The

"multi-instance" setup with local Instances configured with a pre-2014 of Polarion still works with Polarion 2014 and newer release without any changes in the configuration. However, it is no longer possible to create new local instances.

If you want to configure a clustered Instance from any of your local Instances, then you need to migrate the whole setup to the new Multiple Stand-alone Instances setup, where Instances always run on a dedicated machine.

### 5.1 Common terms

- **[INSTALL]** - The root directory of your current installation. This would typically be `C:\Polarion` on Windows or `/opt/polarion` on Linux.
- **[APACHE]** - Apache configuration directory. On Linux should be `/etc/httpd/conf.d/` and on Windows `C:\Polarion\bundled\apache\conf\extra`

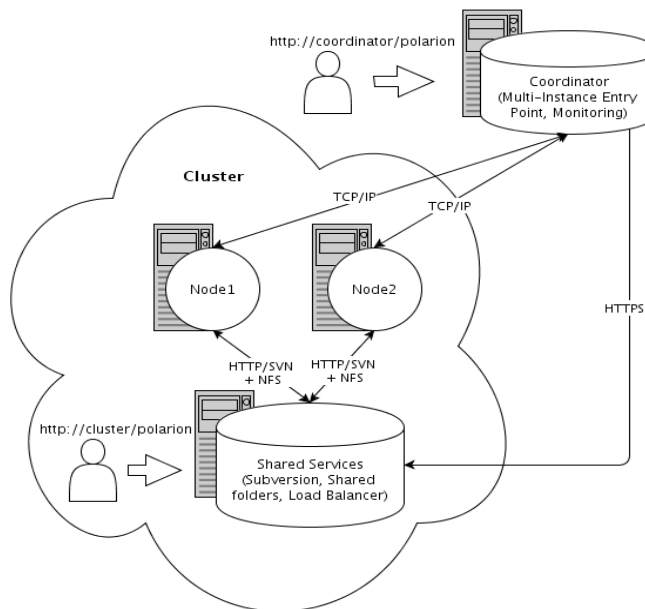
### 5.2 Setting up a Cluster from New Installations

This section describes how to set up a simple Cluster with two Nodes from new/clean Polarion installations. These machines must be running the same version of Polarion.

Prerequisites - 4 machines (virtual or physical):

1. Coordinator (<http://coordinator.yourdomain.com>)
2. Node1 (<http://node1.yourdomain.com>)
3. Node2 (<http://node2.yourdomain.com>)
4. Shared Services (<http://cluster.yourdomain.com>)

#### Deployment diagram:



Start by installing the same version of Polarion on each of the machines: Coordinator, Node1, Node2 and Shared Services.

Note that different third-party software is required on individual machines:

- On the Nodes - Java, Apache HTTP Server, and PostgreSQL
- On the Coordinator - Java, and Apache HTTP Server

- On the Shared Services - Apache HTTP Server, and Subversion

The easiest way is to use the standard installation procedure to install all dependencies and eventually uninstall the software that is not needed, if you need to save space on the storage. This software is bundled in Windows distributions, and is already present on most Linux installations. Refer to the Polarion Installation Guide for your preferred operating system for complete installation instructions. (This PDF guide is bundled in all Polarion distributions and is also available for download on the Polarion Software web site at: <https://www.polarion.com/guides-and-manuals>.)

Instantiation of a local Subversion repository must be done only on the Shared Services machine, as it is the only repository that will actually be used. **Polarion should not be started immediately after installation**, as further changes in configuration are required in order to set up the cluster.

Because Coordinator serves as a license hub for all the nodes and instances connected to it, you do not need to activate any licenses on the nodes.

The next steps in the process assume that Polarion is successfully installed on each node. There are specific next steps are specific for each machine.

### 5.2.1 Configuring the Cluster's Coordinator

*To configure the Coordinator:*

1. Stop Polarion.
2. Make a backup of the original `polarion.properties` file.
3. Replace `polarion.properties` using the Coordinator template file provided in `[INSTALL]/polarion/install` folder: `polarion.properties.template.coordinator`. (Replace the original `polarion.properties` file with a copy of the template file, renaming that as `polarion.properties`).
4. Make the following changes in the template-derived properties file, following comments from the template:
  - Specify `base.url` appropriately for the machine. Must be FQDN or IP address.
  - Set the same value for `ajp13-port` as in the original `polarion.properties` file.
  - Set the same value for `controlPort` as in the original `polarion.properties` file.
  - Specify `controlHostname` appropriately for the machine.
5. (Optional) Uncomment two properties about load balancer credentials if Apache Load Balancer is protected using basic authentication according to step 3 in chapter 5.2.2 (username and password).
  - The default setup uses the same credentials as the svn repository.
6. (Optional) Change the ZooKeeper port if the default port specified is not appropriate or blocked by firewall policy.
7. (Optional) To disable the unused SVN repositories on the nodes remove the file `polarionSVN.conf` from the apache configuration directory and restart Apache.
  - apache configuration directory on Linux should be: `/etc/httpd/conf.d/`
  - apache configuration directory on Windows should be: `C:\Polarion\bundled\apache\conf\extra`
8. (Windows only) You must make sure that the polarion service is started with credentials of a domain user created for polarion - the same user for all polarion installations.
9. Start Polarion.

Below is the configured `polarion.properties` file for Coordinator from the steps above:

```
com.polarion.application=polarion.coordinator
base.url=http://coordinator.yourdomain.com
TomcatService.ajp13-port=8889

# Control port and host name for shutdown requests
controlPort=8887
controlHostname=coordinator.yourdomain.com

# Credentials used to connect to the load balancer, if authentication is enabled
# Replace #ClusterId# with the id of the cluster.
#com.polarion.cluster.#ClusterId#.loadBalancer.user=
#com.polarion.cluster.#ClusterId#.loadBalancer.password=

# Port to connect to zookeeper
com.polarion.zookeeper.port=2181
```

### 5.2.1.1 License Deployment on Coordinator

**Polarion 2015 and later:** Activate Polarion license using the Polarion Activation screen on the Coordinator.

Accessing `http://coordinator.yourdomain.com/polarion` or login screen of any Node or Instance will redirect you automatically to the Polarion Activation screen. For more information see [Activation Help](#).

**Polarion 2014 and earlier:** Make sure that the correct license file is placed in the `license` folder prior to starting the server:

- on Linux: `/opt/polarion/polarion/license/`
- on Windows: `C:\Polarion\polarion\license\`

A Cluster's license is activated in the same way as a single instance (described in the Polarion Installation Guide documentation). The activation application runs on the Coordinator machine and instructs the user how to activate on-line or off-line. Users accessing any entry point and the login screens of individual nodes and instances are redirected to the activation page on Coordinator until activation is complete. Nodes and instances can start even if Polarion is not activated, but users cannot log in.

### 5.2.2 Configuring the Cluster's Shared Services

1. Stop Polarion server.
2. Uninstall the Polarion service.
  - On Linux, run this script: `/opt/polarion/bin/uninstall_polarion_service.sh`
  - On Windows, run: `C:\Polarion\polarion\service.bat -uninstall`
3. Configure Load Balancer in Apache using the example template file provided in `[INSTALL]/polarion/install` folder: `loadbalancer.conf.apache24.template` (for Apache 2.4) or `loadbalancer.conf.apache22.template` (for Apache 2.2). Copy it to `[APACHE]` directory and rename it to `loadbalancer.conf`
  - Basic authentication is configured in the template and you need to check the correct location for the `AuthUserFile`.
4. (Windows only) Make sure that `loadbalancer.conf` is included in `httpd.conf`:

```
# Polarion
Include conf/extra/loadbalancer.conf
Include conf/extra/polarion*.conf
```

5. (Windows only) Comment out or remove the following lines from `polarion.conf`:

```
ProxyPass /polarion ajp://127.0.0.1:8889/polarion timeout=600
ProxyPassReverse /polarion ajp://127.0.0.1:8889/polarion
```

6. Restart Apache.

7. Make changes in the template file, following the comments provided:

- Change path for `passwd` appropriately for this machine.
  - On Linux it will be `/opt/polarion/data/svn/passwd`.
  - On Windows it will be `C:\Polarion\data\svn\passwd`
- Adjust `BalancerMembers` to point to the address of each node.
- Adjust `ProxyPassReverse` to point to the address of each node
- (Optional) Uncomment logging directives if you want enable logging for Load Balancer.

8. Set up the shared folder:

- On Linux machines, we recommend NFSv4 protocol for sharing
- On Windows machines you can use CIFS/Samba share for sharing. It must be shared for the same domain user that is used for running all polarion installations in the cluster. The user needs All permissions for the share.
- Data sharing for different Operating systems and protocols is covered in the chapter on **Configuration of Shared Data**.

9. Make backup of the original `polarion.properties` file on this machine.

10. Modify `polarion.properties`:

- The original properties from the clean installation must be preserved. These properties will be shared between nodes in the cluster, so everything that is common to nodes should be there.
- Add property: `com.polarion.zookeeper=coordinator.yourdomain.com:2181`
- Add property: `com.polarion.clusterId=cluster1`
- Add property: `com.polarion.clusterLabel=Main Cluster`
- Add property: `com.polarion.clusterDescription=Description of Main Cluster`
- Add property: `com.polarion.loadBalancer=http://cluster.yourdomain.com/balancer-manager`
- Modify property: `svn.access.file=${com.polarion.shared}/data/svn/access`
- Modify property: `svn.passwd.file=${com.polarion.shared}/data/svn/passwd`
- Modify property: `polarion.build.default.deploy.repository.url=file://${com.polarion.shared}/data/shared-maven-repo`
- Comment out property: `repoSystem`
- Comment out property: `com.polarion.platform.internalPG`

**Note:** this URL must point to the Apache Load Balancer Manager URL. Domain is machine-specific and will be used as the entry point for this cluster.

**Note:** Property `com.polarion.platform.internalPG` has to be present in all nodes `polarion.properties` file

Below is the configured `polarion.properties` for Cluster - Shared Services (this properties file will be used by each node in the cluster).

**NOTE:** You should not use "file" protocol in the `repoSystem` property while using a cluster setup for performance reasons. Either comment out, remove the line, or else set up a "svn" server to use with this property. If you still want to use the "file" protocol, you need to point it to the shared repository.

```
# Newly added properties to original file
com.polarion.zookeeper=coordinator.yourdomain.com:2181
com.polarion.clusterId=cluster1
com.polarion.clusterLabel=Main Cluster
com.polarion.clusterDescription=Description of Main Cluster
com.polarion.loadBalancer=http://cluster.yourdomain.com/balancer-manager

# Modified properties
#repoSystem=...
svn.access.file=${com.polarion.shared}/data/svn/access
svn.passwd.file=${com.polarion.shared}/data/svn/passwd
polarion.build.default.deploy.repository.url=file://${com.polarion.shared}/data/shared-maven-repo

# List of properties from original file
repo=...
etc..
```

## 5.2.3 Configuring the Cluster's Nodes

**IMPORTANT:** The following steps must be performed for each Node in the Cluster.

1. Stop Polarion server.
2. Make a backup of the original `polarion.properties` file.
3. Replace `polarion.properties` using the example template file provided for nodes in  
[INSTALL]/polarion/install folder: `polarion.properties.template.node`
4. Make sure that the shared folder is mounted on this machine on the recommended path:
  - shared folder on Linux should be in `/opt/polarion/shared`
  - shared folder on Windows is accessed directly as `\\<shared_services_host>\Polarion`
5. Make changes in template file following the comments provided:
  - Set `com.polarion.shared` to point to the mounted shared folder (on Linux it should be `/opt/polarion/shared`, on Windows it should be `\\<shared_services_host>\Polarion`)
  - Set the same value for `ajp13-port` as in the original `polarion.properties` file.
  - Set the same value for `controlPort` as in the original `polarion.properties` file
  - Set value for `controlHostname` to `node1.yourdomain.com` or `node2.yourdomain.com` (depending on which node you are configuring).
  - Set the value for `com.polarion.loadBalancer.workerUrl` to the specific node in cluster so that the load balancer knows the URL of the node.
  - Set a value in the `calc.base.url` property to a specific node in the Cluster. It must point to the specific node, otherwise calculation will fail. It is the same as `workerUrl`.  
(Eg. `calc.base.url=http://node1.yourdomain.com`).
  - Add property `com.polarion.platform.internalPG` (with value) from the Cluster's shared services where should be commented out
6. (Optional) To disable the unused SVN repositories on the nodes remove the file `polarionSVN.conf` from the Apache configuration directory and restart Apache.
  - Apache configuration directory on Linux should be `/etc/httpd/conf.d/`
  - Apache configuration directory on Windows should be `C:\Polarion\bundled\apache\conf\extra`
7. (Windows only) Make sure that the Polarion service is started using credentials of a domain user created for Polarion - the same user should be used for all Polarion instances.
8. Start Polarion.

Below is the configured `polarion.properties` file for Node1. (It will be the same for second or third Nodes, except that URLs must be changed accordingly.)

```
# Shared folder between the machines that make up the cluster
# default Linux: com.polarion.shared=/opt/polarion/shared
#default Windows: com.polarion.shared=\\<shared_services_host>Polarion
com.polarion.shared=/opt/polarion/shared
com.polarion.nodeId=node1
TomcatService.ajpl3-port=8889

#Url of node in load balancer
com.polarion.loadBalancer.workerUrl=http://node1.yourdomain.com

# Control port and host name for shutdown requests
controlPort=8887
controlHostname=node1.yourdomain.com

#Node-specific url of the node.
#It is used in calculations to access Polarion via web services
#calc.base.url=http://example-node
calc.base.url= http://node1.yourdomain.com

#Postgres database connection
com.polarion.platform.internalPG=polarion:passwordForDatabase@localhost:5433
```

You have now configured an entire Cluster from new/clean installations of Polarion.

Your cluster will be accessible on: <http://cluster.yourdomain.com>

Server Monitoring is accessible on: <http://coordinator.yourdomain.com/polarion/monitoring>

Apache Load Balancer Manager is accessible on: <http://cluster.yourdomain.com/balancer-manager>

#### 5.2.3.1 Synchronizing Time on Cluster Nodes

**IMPORTANT:** Time must be synchronized on each node in the cluster on the OS level by a system administrator. Ideally this should be automated sync via NTP. If time is not synchronized, users will see different times on each node, scheduled jobs may appear to start off schedule and the Monitor will incorrectly order jobs by time.

#### 5.2.4 Configuring the Cluster's Activation Application

Beginning with version 2015, Polarion includes an activation application that makes it possible to install or update a license during runtime of the Polarion server, without need to copy the license file manually to the target machine. Access to this application is NOT protected initially by user name and password. For production use it is highly recommended to secure access to this application directly in the Apache configuration. It is necessary to perform this configuration only on the coordinator server.

In version 2015 installations there is a template Apache configuration file in the Polarion installation folder:  
`/polarion/polarion/install/polarion.activation.conf.template`

To ensure that a user name and password is requested when accessing the activation application (`/polarion/activate/online` and `/polarion/activate/offline`), copy this file to the Apache configuration folder. On Linux usually `/etc/httpd/conf.d/`.

On Windows, usually `C:\Polarion\bundled\apache\conf\extra\`.

After copying the file, rename it to remove the extension ".template". Then open the file in any text editor and modify it according to the instruction comments provided.

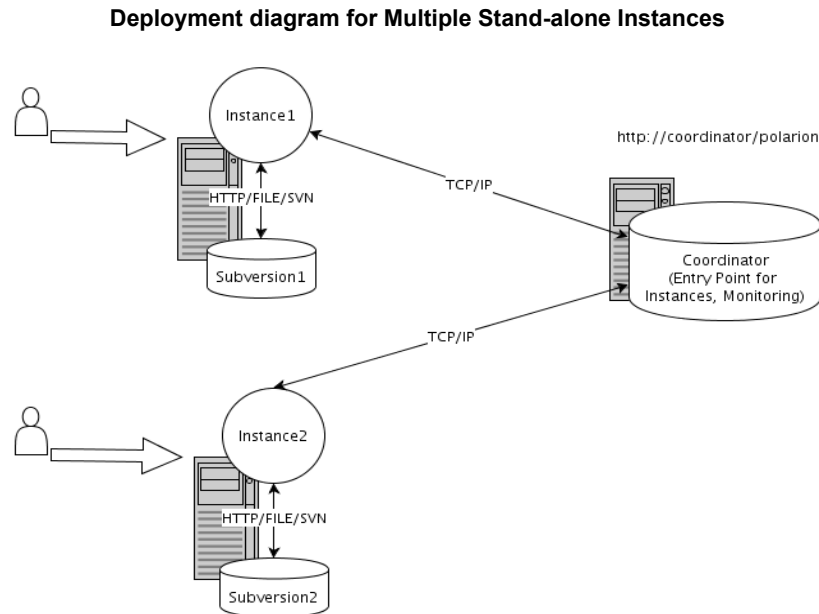
The template configuration is prepared for both users file authentication (like Polarion uses for Subversion by default, with user passwords data in a file) and for authentication against an LDAP server.

### 5.3 Setting Up With Multiple Stand-alone Instances

You can set up the Multiple Stand-alone Instances configuration using the Coordinator for license management.

Three machines (virtual or psychical) are required for this setup:

1. Coordinator (<http://coordinator.yourdomain.com>)
2. Stand-alone Instance1 (<http://instance1.yourdomain.com>)
3. Stand-alone Instance2 (<http://instance2.yourdomain.com>)



Start by installing the **same version** of Polarion on each of the 3 machines: Coordinator, Instance1, and Instance2.

Note that different third-party software is required on individual machines:

- On the Instances - Java, Apache HTTP Server, Subversion, and PostgreSQL
- On the Coordinator - Java, and Apache HTTP Server

The easiest way is to use the standard installation procedure to install all dependencies and eventually uninstall the software that is not needed, if you need to save space on the storage. This software is bundled in Polarion distributions for Windows, and is already present on most Linux installations. Refer to the Polarion Installation Guide document bundled with Polarion distributions and available on the Polarion Software web site at: <https://www.polarion.com/guides-and-manuals>.

The next sections assume that Polarion is successfully installed using the standard installation and running on each machine. There are specific next steps that need to be performed on each machine.

### 5.3.1 Configuring the Coordinator for Multiple Stand-alone Instances Setup

Configuration of the Coordinator is exactly same as for the Cluster setup described in chapter 5.2 *Setting up a Cluster from New Installations*, section 5.2.1 *Configuring the Cluster's Coordinator*. Proceed to configure the Coordinator for this setup as described there.

The next section on Instances configuration will refer to this Coordinator machine (<http://coordinator.yourdomain.com>) and assumes that the Coordinator is configured and running.

### 5.3.2 Configuring Instance1

*On the machine hosting the Polarion installation for Instance1:*

1. Stop Polarion.
2. Make backup of the original `polarion.properties` file.
3. Modify `polarion.properties`:



- All properties in the original file must be preserved.
- Add new property: `com.polarion.zookeeper=coordinator.yourdomain.com:2181`
- Add new property: `com.polarion.clusterId=Cluster1`
- Add new property: `com.polarion.nodeId=Instance1`
- Add new property: `com.polarion.clusterLabel=First Server`
- Add new property: `com.polarion.clusterDescription=Description of first Server`

#### 4. Start Polarion.

Below is the configured `polarion.properties` file for Instance1:

```
# Newly added properties to original file
com.polarion.zookeeper=coordinator.yourdomain.com:2181
# com.polarion.clusterId - is it identifier on coordinator
# (instance displays as independent cluster)
com.polarion.clusterId=Cluster1
com.polarion.nodeId=Instance1
com.polarion.clusterLabel=First Server
com.polarion.clusterDescription=Description of first Server

# List of properties from original file
repo=...
repoSystem=...
etc..
```

### 5.3.3 Configuring Instance2

On the machine hosting the Polarion installation for Instance2:

1. Stop Polarion.
2. Make a backup of the original `polarion.properties` file.
3. Modify `polarion.properties`:
  - All properties from original file must be preserved
  - Add new property: `com.polarion.zookeeper=coordinator.yourdomain.com:2181`
  - Add new property: `com.polarion.clusterId=Cluster2`
  - Add new property: `com.polarion.nodeId=Instance2`
  - Add new property: `com.polarion.clusterLabel=Second Server`
  - Add new property: `com.polarion.clusterDescription=Description of second Server`
4. Start Polarion.

Below is the configured `polarion.properties` file for instance2:

```
# Newly added properties to original file
com.polarion.zookeeper=coordinator.yourdomain.com:2181
# com.polarion.clusterId - is it identifier on coordinator
# (instance displays as independent cluster)
com.polarion.clusterId=Cluster2
com.polarion.nodeId=Instance2
com.polarion.clusterLabel=Second Server
com.polarion.clusterDescription=Description of second Server

# List of properties from original file
repo=...
repoSystem=...
etc..
```

The configuration is quite similar to the Cluster setup. The difference is that there is no Load Balancer or Shared

Services. Each Instance is autonomous, a "stand-alone" Polarion installation with its own SVN repository. Individual Instances have nothing to do with other Instances in the Multiple Stand-alone Instances setup. However, users can easily switch between the Instances by accessing the Entry Point on the Coordinator. You can also monitor availability of each Instance using Server Monitoring.

**NOTE:** The Polarion UI and end-user documentation use the term "Server" when referring to what we term "Instance" for administrators. For example, the UI provides end users the possibility to "Change Server". In Administration terms, that is... work on a different Instance.

### 5.3.4 Access URLs for Multiple Stand-alone Instances

Entry Point (all Instances): `http://coordinator.yourdomain.com/polarion`

Server Monitoring: `http://coordinator.yourdomain.com/polarion/monitoring`

Instance1 direct access: `http://instance1.yourdomain.com/polarion`

Instance2 direct access: `http://instance2.yourdomain.com/polarion`

## 5.4 Migrating From a Pre-2014 "Multi-Instance" Installation

Several Polarion versions prior to version 2014 supported a topography of multiple Polarion Instances which was called "Multi-Instance" setup. Instance clustering was not supported. Although existing customer installations of this setup are still usable with version 2014 and later, **the setup is now considered deprecated** in favor of the many improvements delivered in version 2014, and it is generally recommended to change over such existing installations. This chapter provides information and procedures. Customers with a current support and maintenance package may consult Polarion Technical support for assistance with this migration.

The new *Multiple Stand-alone Instances* setup differs from old Multi-Instance setup in the following ways:

- The so-called *Master* is replaced by the Coordinator, which manages a license for all Instances.
- Local *Instances* are not compatible with the new Multiple Stand-alone Instances setup. For backward compatibility, all operations on existing local Instances are still maintained after updating to Polarion 2014 or later. However, you will not be able to create *new* Instances. If you have local Instances configured and wish to update to 2014+ Multiple Stand-alone Instances, these local Instances should be moved to separate machines and then configured later as part of a Multiple Stand-alone Instances setup (see 5.3.3 *Moving Local Instances for the 2014 Multi-Instance Setup*).
- Each Remote Instance will become a non-clustered Instance connected to the Coordinator.
- The Coordinator does not start up the Instances... these must be started individually.

In order to do the migration, you need to update Polarion on the old "Master" and "Remote" Instances to the same version. Then you need to modify the configuration files so that they reflect the new configuration properties.

For example: A pre-2014 setup with 1 Master application and 2 remote Instances will become a 2014 Multiple Stand-alone Instances setup with 1 Coordinator and 2 non-clustered Instances, each Instance hosting a stand-alone installation of Polarion, complete with third-party software and repository.

### 5.4.1 Configuring the Coordinator

To replace the pre-2014 Multi-Instance setup you need to configure the Coordinator. Coordinator is still running on the machine where the Master and Local Instances were running.

Follow the steps described in section 5.2.1 *Configuring the Cluster's Coordinator*, using also the information from `_controller.properties` if needed. For example, `controlPort` and `controlHostname` can be taken from `_controller.properties`.

From this point on, it is assumed that you have the Coordinator configured, running and accessible through following

URL: <http://coordinator.yourdomain.com/polarion>.

#### 5.4.2 Migrating a Remote Instance to a Non-clustered Stand-alone Instance

The following changes need to be done on each Remote Instance.

1. Stop the Instance and update it to the latest Polarion version (2014 or later) in the usual way.
2. Make backup of the original `polarion.properties` file.
3. Modify `polarion.properties`:
  - All properties from original file must be preserved
  - Add new property: `com.polarion.zookeeper=coordinator.yourdomain.com:2181`
  - Add new property: `com.polarion.clusterId=OldRemoteInstanceId`
  - Add new property: `com.polarion.nodeId=OldRemoteInstanceId-node1`
  - Add new property: `com.polarion.clusterLabel=Old Remote Instance Label`
  - Add new property: `com.polarion.clusterDescription=Description of the old remote instance`
4. If you have any properties configured in `instanceid.properties`, they should be moved into `polarion.properties` otherwise they will be ignored.
5. Start Polarion.

Below is an example of a `polarion.properties` file for a migrated remote instance (the instance ID is *instance1*):

```
# Newly added properties to original file
com.polarion.zookeeper=coordinator.yourdomain.com:2181
com.polarion.clusterId=instance1
com.polarion.nodeId=node1
com.polarion.clusterLabel=Remote instance - Instance1
com.polarion.clusterDescription=Description of the remote instance

# List of properties from original file
repo=...
repoSystem=...
etc..
```

##### 5.4.2.1 Checking the Migration

To check that the migration was successful, go to <http://coordinator.yourdomain.com/polarion> and connect to the Instances.

Entry Point URL: <http://coordinator.yourdomain.com/polarion>

Server Monitoring URL: <http://coordinator.yourdomain.com/polarion/monitoring>

Each Instance can be still directly accessed through its URL: e.g. <http://instance1.yourdomain.com/polarion>

The old configuration files for the pre-2014 Multi-Instance setup (from `[polarion_installation]/configuration/multi-instance/*`) will become obsolete.

#### 5.4.3 Moving Local Instances for the Multiple Stand-alone Instances Setup

Moving a Local Instance means moving the existing repository and the configuration files to a new Polarion installation.

This step is only required if some of the Instances are configured as Cluster. If no Cluster is needed, the Local Instances will still work as they did before in the old Multi-instance setup with the same configuration.

##### Linux paths:

- `polarion.properties` file: `/opt/polarion/etc`
- Repository folder: `/opt/polarion/data/svn` OR `/opt/polarion/data/multi-instance/instanceId/svn`

### Windows paths:

- `polarion.properties` file: `C:\Polarion\polarion\configuration`
- Repository folder: `C:\Polarion\data\svn` OR `C:\Polarion\data\multi-instance\instanceId\svn`

### To move a Local Instance to a new machine:

1. Install Polarion on the new machine. Make sure it can be started correctly, then stop it and keep it stopped for the next steps.
2. In the new installation location: make a backup of the repository folder. (This copy will subsequently be referred to as folder. (This copy will subsequently be referred to as `svn_backup`)
3. In the new installation location: make a backup of the `polarion.properties` file. (This copy will subsequently be referred to as `polarion.properties_backup`).
4. Copy the repository folder from the old Instance to the new machine at location:  
Linux: `/opt/polarion/data`  
Windows: `C:\Polarion\polarion\data`
5. Copy the `polarion.properties` from the old instance to the same location on the new machine (see path references above).
6. Start Polarion. You should have all the data from the old instance.
7. After successful startup, you can delete `svn_backup` and `polarion.properties_backup`

At this point you have a clean installation of the latest Polarion version holding the data and configuration of the old Instance. You can configure this Instance as part of a Multi-Instance setup following the steps described in [5.2 Setting Up a Clean Multi-Instance Installation](#).

### 5.5 Updating a Multiple Stand-alone Instances or Cluster setup

When updating either setup, it is important that you first **shut down all the instances and the coordinator**. After that, you can use the [update distribution](#) to update the machines in the setup (see steps below).

**IMPORTANT:** It is not possible to run a Cluster node with old version of Polarion and at the same time have it connected to the updated Coordinator. Be sure you schedule extended downtime - enough to install the update on all machines and for the necessary reindexing.

#### Update steps:

1. Stop the Polarion service on all nodes, and finally on the Coordinator.
2. Install the update on the coordinator machine and start Polarion in reindex mode.
3. Install the update and start each of the cluster nodes in reindex mode. **IMPORTANT: update the Cluster nodes in sequence**, so that only one of the nodes does an update of the SVN repository. (Updating the shared services server is not required.)

## 6 Configuring Shared Data

This chapter provides steps how to configure shared data on Linux and Windows machines. The setup is different for each platform and all differences are covered in this chapter.

Prerequisites (4 machines, all on same domain):

1. Coordinator (*coordinator.yourdomain.com*)
2. Node1 (*node1.yourdomain.com*)
3. Node2 (*node2.yourdomain.com*)
4. Shared Services (*cluster.yourdomain.com*)

The shared folder has same structure as standard Polarion installation folder, so it is possible to use a Polarion installer to create it:

1. Install Polarion.
2. Uninstall Polarion service and delete folders that are not needed. Only two folders in the Polarion installation are needed for shared data:
  - Linux: `/opt/polarion/etc` and `/opt/polarion/data`
  - Windows: `C:/Polarion/polarion/configuration` and `C:/Polarion/data`

**Note:** Deleting of other (unnecessary) folders is optional. You can optionally leave the installation folder as it is after installation.

The root of the Shared Services is the `polarion` folder.

## 6.1 Linux Configuration

Share the folders among the nodes **NFSv4** protocol. Other protocols (such as SSHFS or NFSv3) have known problems, so they must not be used.

### 6.1.1 NFS Configuration

This section describes an example how to set up folder sharing using the NFS protocol.

*To configure folder sharing using NFS:*

1. Connect to the Shared Services machine (*http://cluster.yourdomain.com*)
2. Edit file `/etc/exports` and add the following lines:
  - Add line: `/opt/polarion node1(rw, sync, no_root_squash, no_subtree_check)`
  - Add line: `/opt/polarion node2(rw, sync, no_root_squash, no_subtree_check)`
3. On the *Node* machines create a folder `/opt/polarion/shared`
4. On the *Node* machines add following line to `/etc/fstab`
  - Add line: `cluster.yourdomain.com:/opt/polarion /opt/polarion/shared nfs defaults 0 0`
5. On all machines run following command:
  - # `/etc/init.d/rpcbind start`
  - # `/etc/init.d/nfs start`
6. On the Shared Services machine run following command:
  - # `exportfs -a`
7. And on the *node* machines mount the shared directory with the command:
  - # `mount -v cluster.yourdomain.com:/opt/polarion /opt/polarion/shared/`
8. Check that **shared** folder appears on each node in their the `/opt/polarion` folder, and make sure that `polarion.properties` on each node points to this location: `/opt/polarion/shared`
9. Make sure that each node has **rw** permissions for the `/opt/polarion/shared` folder, and all nodes create folders and files with the same permissions.

## 6.2 Windows Configuration

This section describes how to make a shared folder on a Windows machine. We recommend using standard Windows sharing on this platform.

## 6.2.1 CIFS/Samba Share Configuration

You need to configure sharing on the Shared Services machine, and all Node machines, starting with the Shared Services machine.

### 6.2.1.1 Shared Services Machine

A simple example on how to create the shared folder using CIFS/Samba.

*To create a shared folder on the Shared Services machine:*

1. Connect to the Shared Services machine (*http://cluster.yourdomain.com*)
2. Open File Explorer
3. Right-click on the `C:/Polarion` folder
4. Select **Properties**.
5. Select tab **Sharing**.
6. Click the **Share...** button.
7. Set shared user as the same domain user for all polarion installations in the cluster. The user needs full permissions to the folder.
8. After you have done with sharing options, click on **Share**, and then on **Done**.

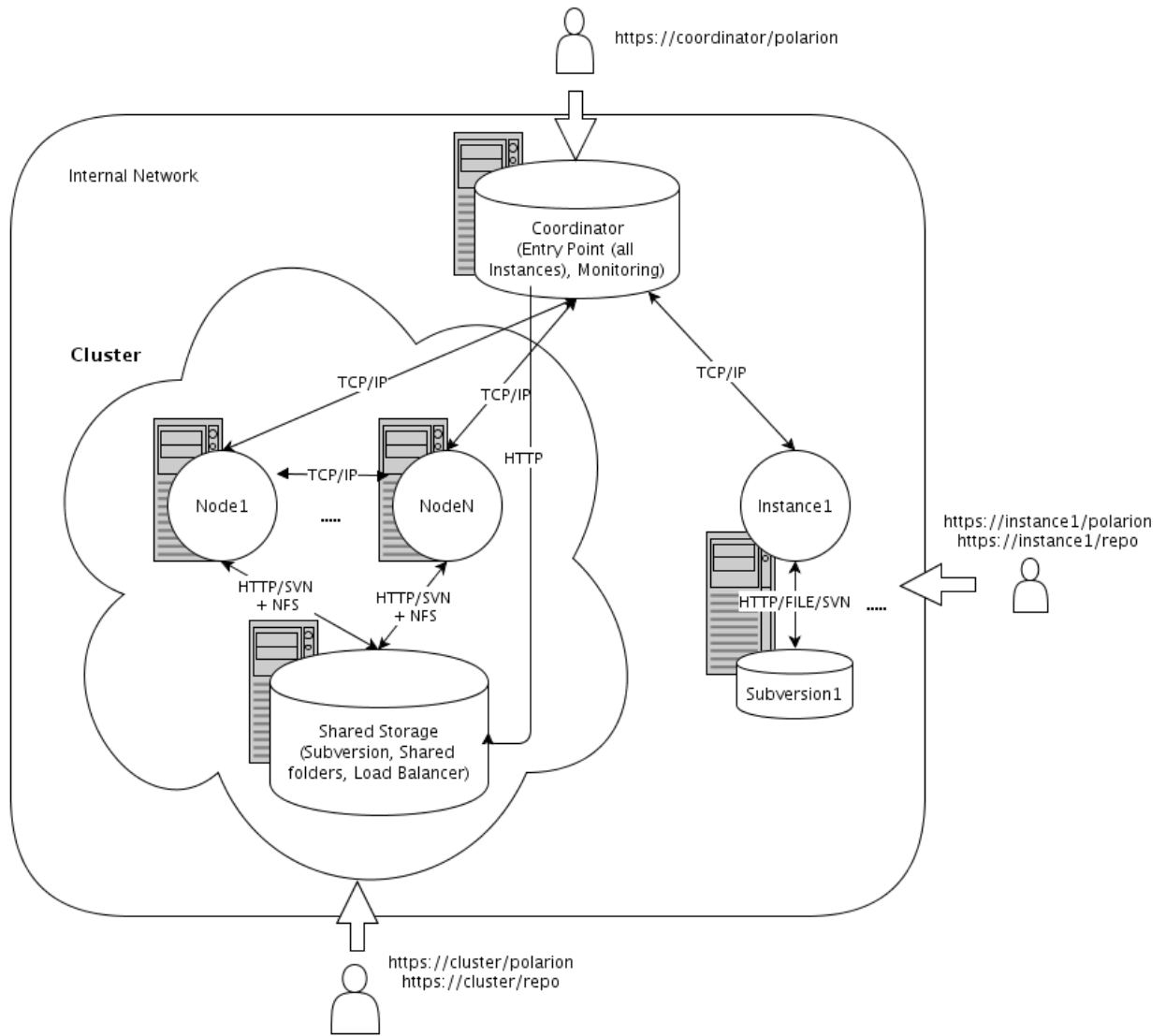
### 6.2.1.2 Node Machines

*To connect shared folder on each Node:*

1. Connect to each Node (*http://node1.yourdomain.com* and *http://node2.yourdomain.com*)
2. Open File Explorer
3. In the left panel, right-click on **Computer**.
4. Map a network drive. Using the credentials of the polarion domain user.
  - Folder should be: `\\cluster.yourdomain.com\polarion`
5. Edit the `polarion.properties` file accordingly and specify the path to the shared folder.
  - Property `com.polarion.shared` must point to this mapped drive.

## 7 Security Options

The recommended setup is to use encrypted communication between the outside world and the internal network with servers of the Multiple Stand-alone instances setup (as shown in the figure below). This is also optimal from a performance point of view. Communication inside the local network can optionally be encrypted as well (except for the folders shared using NFS). More details about this are provided in the section *Advanced Security Options*.



In the sections below it will be noted that https access should be set up in an Apache server. The details about how to actually configure this are out of scope of this guide, however you can see some hints and references in Polarion Help.

### 7.1 Recommended Security Options

Service	Security Settings
Entry point (Coordinator)	The entry point, where users can select Polarion server, should be configured for https access in Apache, so that end users will access e.g. <code>https://coordinator.mycompany.com/polarion</code> .  <b>Additional steps:</b> Remember to update <code>base.url</code> in Polarion properties.
Server monitoring (Coordinator)	The same as above for the server monitoring page, e.g. <code>https://coordinator.mycompany.com/polarion/monitoring</code> . This will be usually done by the same configuration as the entry point.
Stand-alone Instance Polarion	Standard https setup as for simple stand-alone installation, so that the instance can be accessed as e.g. <code>https://instance1.mycompany.com/polarion</code> .  If the Subversion repository is accessed by end users, it should be configured for https access as well.

	<p><b>Additional steps:</b>  Remember to update <i>base.url</i> in Polarion properties.</p>
Clustered Instance Polarion	<p>Standard https setup in Apache for the Load Balancer, so that the clustered instance can be accessed as e.g. <i>https://instance2.mycompany.com/polarion</i>. If the Subversion repository is accessed by end users, it should be configured for https access as well.</p> <p><b>Additional steps:</b></p> <ol style="list-style-type: none"> <li>1) Note that in this case it is necessary to set Polarion property <i>wikiProtocolSchema=https</i> in the shared cluster properties (<i>/opt/polarion/etc/polarion.properties</i> on the Shared Services).</li> <li>2) Remember to update <i>base.url</i> in shared cluster properties.</li> </ol>

## 7.2 Advanced Security Options

If desired the internal network communication among the servers comprising the Multiple Stand-alone Instances setup can be encrypted as well.

Service	Security Settings
Load Balancing	<p>Load balancing communication between load balancer and the workers (clustered instances) can be done via https if desired. It is necessary to set up https access on Coordinator and all cluster Instances (as for simple installation), and then configure the load balancer to use the https worker URLs. You can use the same wildcard certificate on all servers.</p> <p><b>Additional steps:</b></p> <ol style="list-style-type: none"> <li>1) It is necessary to switch on SSL proxy engine using <i>SSLProxyEngine on</i> in the Apache configuration.</li> <li>2) It is necessary to set Polarion property <i>wikiProtocolSchema=https</i> in the shared cluster properties (<i>/opt/polarion/etc/polarion.properties</i> on the Shared Services).</li> <li>3) Remember to update <i>base.url</i> in shared cluster properties.</li> </ol> <p>Note that by default, Apache does not verify the certificate of the workers. To switch it on, set <i>SSLProxyVerify=require</i> and you might also need to set <i>SSLProxyCACertificatePath</i> or other directives. See <a href="http://httpd.apache.org/docs/2.2/mod/mod_ssl.html">mod_ssl documentation</a> (<a href="http://httpd.apache.org/docs/2.2/mod/mod_ssl.html">http://httpd.apache.org/docs/2.2/mod/mod_ssl.html</a>) for more details.</p>
Load Balancer Management	<p>By default, the Coordinator manages the load balancer. For example, it switches off the worker if a Polarion cluster node disconnects from the cluster. This management is done using the http/https URL provided by the shared cluster property <i>com.polarion.loadBalancer</i>. The load balancer manager is a web application provided by Apache, and it can be configured for https access (on Shared Services).</p> <p><b>Additional steps:</b></p> <ol style="list-style-type: none"> <li>1) Remember to update Polarion property <i>com.polarion.loadBalancer</i> in shared cluster properties.</li> <li>2) It might be necessary to install a trusted certificate authority to Java trust store, especially if a self-signed certificate is used.</li> </ol>
Subversion Repository	<p>For best performance, cluster nodes should access the shared repository by the system user using the SVN protocol (<i>repoSystem=svn://...</i>). This makes it necessary to open remote access to <i>svnserve</i> running on the Shared Services machine. This communication is not encrypted.</p> <p>To enhance security you may want to consider establishing a secure channel using, for example, <a href="https://www.stunnel.org/">stunnel</a> (<a href="https://www.stunnel.org/">https://www.stunnel.org/</a>). The idea is: instead of...</p> <pre>repoSystem=svn://SHARED_SERVICES_HOST/opt/polarion/data/svn/repo</pre> <p>use...</p> <pre>repoSystem=svn://localhost/opt/polarion/data/svn/repo</pre>



	...on the cluster node and connect localhost:3690 to SHARED_SERVICES_HOST:3690 by a secured channel.
Cluster Coordination	Instances and Nodes in the Multiple Stand-alone Instances setup communicate with the Coordinator machine. This communication is not encrypted. It can be secured using, for example, <i>stunnel</i> in a very similar way to that described above. Cluster instances also communicate directly with other cluster instances using TCP socket (on the <i>controPort</i> ). This communication is encrypted internally.

### 7.3 Authentication for Server Monitoring

After initial installation of a cluster, the *Server Monitoring* page is available on the Coordinator. On this page, administrators can view and access all the configured nodes (servers), and manage the load balancer. The access URL for this page has this form: `http://coordinator.yourdomain.com/polarion/monitoring`.

The page does not require any authentication. However, authentication *is* recommended and you can configure basic authentication via the standard way in Apache, using one of the following directives (assuming the password file is either `/opt/polarion/etc/passwd` or `C:/Polarion/data/svn/passwd` - use the one applicable for your OS):

Apache 2.4 and newer:

```
<Location /polarion/monitoring>
  Require all denied
  AuthType Basic
  AuthName "Monitoring"
  AuthUserFile "C:/Polarion/data/svn/passwd"
  Require valid-user
</Location>
```

Apache 2.2 and older:

```
<Location /polarion/monitoring>
  Order Deny,Allow
  Deny from all
  AuthType Basic
  AuthName "Monitoring"
  AuthUserFile "/opt/polarion/etc/passwd"
  Satisfy Any
  require valid-user
</Location>
```

## 8 Notes

- Extensions are not shared among the Nodes in a Cluster. Each node has its own independent extensions folder (e.g. `/opt/polarion/polarion/extensions`). Therefore, an extension can be installed on some specific Node(s) in a cluster. However, in most cases you will want to install extensions on *all* Nodes. Only in the case of some special kind of extension would you *not* install it on all Nodes.
- **Scheduled jobs** should be reviewed for a cluster, and convenient Node selectors (i.e. the *node* attribute of `<job>` elements) should be specified depending on the nature of the job. The following default jobs should have `node="*"` specified: *Index Checker*, *Suspend DB History Creator*, *Resume DB History Creator*.
- **Diagnostics:** Polarion comes with a self-diagnostic utility *Polarion Diagnostic Tool* (PDT), which can run comprehensive diagnostic tests and communicate the results to Polarion's technical support team. PDT checks if Polarion is running in a cluster and gathers configurations from shared folders. The utility is located in the *diagtool* folder under the root of any installed Polarion instance, which also contains documentation for its use.

## 9 Troubleshooting

- (Linux) It is recommended to disable SELinux, if it is used.
- (Windows) Disabling firewall on enterprise editions of Windows also disables crucial network services.
- After encountering problems with activities (e.g. *org.apache.lucene.index.IndexNotFoundException: no segments\* file found in MMapDirectory@/opt/polarion/shared/data/workspace/polarion-data/index/Activities*), it is necessary to manually delete index of activities from the Shared Folder, by default: */opt/polarion/shared/data/workspace/polarion-data/index/Activities* and restart one of the nodes (so that new empty index is created).

## 10 Appendix: Polarion Instance Architecture

