Table of Contents

Use Cases of WLCG EGEE Pre Production

Scope of this document

This page describes use cases, workflows and high-level interactions between sites, operators and Virtual Organisation in the context of the WLCG/EGEE preproduction.

It is addressed to VO managers, ROC managers, site managers in the pre-production orbit, and members of the middleware certification and release teams (SA1/SA3).

Since this document provides guidelines for the communication among different partners involved at various titles in the pre-production activity, its final version, as well as each major revision, must be subject to the approval of representatives of:

- VOs interested/involved in the pre-production activity.
- Operation teams (SA1)
- Certification and Release teams (SA3)
- EGEE ROC Managers

Minor changes, namely those dealing with purely technical details, may be decided by the Pre-Production Coordination and notified to the concerned partners.

Mission of the EGEE Pre-Production Service

The EGEE Pre-Production provides access to grid services in preview to WLCG/EGEE interested users, in order to test, evaluate and give feedback to changes and new features of the middleware.

In addition to that, the pre-production extends the middleware certification activity, helping to evaluate deployment procedures, [inter]operability and basic functionality of the software against operational scenarios reflecting real production conditions

This document covers the first of these aspects, highlighting the four basic use cases of the preproduction as they are perceived by users from the VOs. These cases correspond to the four possible type of software upgrade which users and site administrators can be called to cope with.

Use Cases

The pre-production offers, in general, previews of new middleware functionalities to interested users.

Two major classes of previews are supported :

- Previews of client tools (changes/additions affecting Worker Nodes, User Interfaces, VOBOXes)
- Previews of grid services (a.k.a. *pilot* or *experimental* services)

New versions of both client tools and middleware services are made available and selectable to users from the VOs in the production environment (using the production information system and accessing production resources).

For the clients this happens in a semi-automated way, and users from any supported VOs can make use of the new versions.

Preview instances of grid services, on the contrary, are created only on-demand, meaning that supported user scenarios, scope and goals of each pilot have to be agreed in advance between the interested VOs and the PPS. Every EGEE VOs are welcome to submit proposal for new testing to be done either by direct contact to pps-suppor@cernNOSPAMPLEASE.ch or via one of the weekly WLCG/EGEE operations meeting The WLCG/EGEE Operations Meeting and the WLCG Service Coordination Meeting

In other words, there are no more permanent instances of every services available by default to users as it used to be the case in the previous implementation of the PPS.

Fully backward-compatible client update

Definition: By *backward compatible client update* we mean specifically those clients updates where no variables (e.g. YAIM variables) are to be changed and **no new variables** are added to configuration (*).

A predominant fraction of client updates falls in this category, the notable characteristics of which are:

- New clients are compatible with old servers
- Updates often related to bug fixes   time-to-production is more important for VOs
- Empty set of configuration instructions in release notes   extended test of release notes in PPS not needed

The new clients are distributed, possibly **before the certification is complete**, to collaborating sites **in production**, and made available for the VOs to test. As a preconditions to be distributed, the software must have passed a first round of basic certification tests although the full certification process my be not completed yet. The patches in certification that have reached this level of "maturity" are identified and flagged (*) by SA3.

The installation of new clients at the sites does not affect the existing production instances, so that, the site is still fully functional for the production work.
These "preview" installations inherit the same local settings used by the production clients (e.g. environment settings in profile.d).
There is a dedicated tag in the production information system (using the attribute `GlueHostApplicationSoftwareRunTimeEnvironment`) to announce that a site supports these non-certified releases.
The distribution, installation and publishing of independent versions of clients is handled by a centralised mechanism.

The "live" map of the distribution of new versions over the various sites is available in a web page in the pre-production website. Users from the VOs are able to select the desired version using a particular requirement in the jdl. This feature is available by default only to users submitting **through the gLite WMS**. Jobs submitted bypassing the WMS will use instead the standard "production" version of the clients.

The deployment of a new client version in pre-production starts when the *release manager* (from SA3) decides that a patch in certification (Status="In certification") can be "moved" to this deployment area. As a consequence of this decision, the following actions are triggered and carried-out **in parallel with the standard certification**

1. Creation of the tarball
2. Distribution to the sites (ideally using a SAM job)
3. Local testing at the sites (ideally using a SAM job)
4. Publication of the tag in the information system ideally using the same SAM job and through the *lcg-ManageVOTag* command)

5. Update of the "release bulletin" documenting the versions available at the various sites (this can be done automatically on the PPS web site based on the information extracted from Savannah)
6. General notification to potentially interested subjects (e.g. broadcasts to VOs and ROCs).
7. Personal notification to specifically interested subjects (e.g. the originator(s) of the bug(s)/request(s) fixed by a patch released) with the invitation to verify the provided solution.

Immediately after the deployment of a new client version, a dedicated public channel is made available to users to provide feedback. The feedback provided is taken into account within the parallel certification process and eventually it is propagated and summarised in the release notes. In particular the release notes mention explicitly the case in which **no feedback** was provided.

After the parallel certification and preview phases are completed, if the release notes are confirmed not to contain any special configuration information, and limited to the platforms on which an installation test was done in certification, the release is deployed in production with no further deployment testing.

In the next sections the supported update/addition schemes are detailed.

Non-backward-compatible client update

Definition: client updates where interventions on the environment or extra configuration is needed.

The relevant characteristics of this category of updates are:

- Updates often related to new features   time-to-production must comply to VO schedules but not dramatic from the service point of view
- Configuration instructions may be needed in release notes   pre-deployment test of release notes is needed
- New clients are still compatible with old servers. The case of incompatibility is dealt with together with the case of non backward compatible server update

From the VO perspective the way this use case works is exactly the same as the previous one, with some considerations needed.

A local configuration actions are in general needed, the new clients cannot be simply "pushed" to the sites by the deployment team as in the previous case. So a longer elapsed time for deployment has to be expected.

In fact the operations to be performed after the decision to deploy the client in preview is taken are:

1. *Creation of the tarball*
2. Distribution of the client to a number of selected production sites (PP "Silver" partners) (ideally using a SAM job)
3. Local configuration of the clients at the sites
4. *Local testing at the sites (ideally using a SAM job)*
5. *Publication of the tag in the information system (ideally using the same SAM job and through the _lcg-ManageVOTag* command)_
6. *Update of the "release bulletin" documenting the versions available at the various sites (this can be done automatically on the PPS web site based on the information extracted from Savannah)*
7. *General notification to potentially interested subjects (e.g. broadcasts to VOs and ROCs)*
8. *Personal notification to specifically interested subjects (e.g. the originator(s) of the bug(s)/request(s) fixed by a patch released) with the invitation to verify the provided solution*

The pilot is meant to allow the functionality testing, whereas the deployment test in the PPS infrastructure, based on YAIM and release notes is focused on several deployment scenarios (OS, architectures). The two activities run in parallel.

Backward-compatible server update

Definition

With *backward-compatible (BC) server update* we mean updates that are compatible with the existing clients and don't introduce new functionalities. No changes in database schema. Backward compatible updates can in general be rolled back with not relevant information loss. A further distinction in this category is done between *minor* and *major* service updates

Minor: The following conditions have to be *all* true in order to consider an update as "minor"

- no new configuration parameters anywhere (neither in YAIM nor into component-specific configuration files)
- less than 2 "major" plus 5 "normal" bug fixes (according to the severity assigned by the EMT).
- the changes introduces correspond to not more than 2 man-days of programming (this assertion has to be validated by the release manager via a specific attribute in Savannah)
- not significant operational changes are introduced for the service administrators

Major: when any of the above conditions is false.

General policies

Pilot services (aka experimental services) are eventually set-up and run in production upon agreement of the concerned parts (VO, development, certification and operation teams). The purpose of the pilot is to speed-up the process of delivering to production a fully certified and functional service. In this view it is recommendable to set up and upgrade the pilot services using only certified software. Exceptions to this rule may be decided by the concerned parts for justified opportunity reasons. The use of non-certified software in the pilots has however to be justified, documented and recorded in order to safe-guard the reproducibility of the working environment and the integrity of the future releases.

In both major and minor cases, pilot services are set-up and run in production by a number of selected partner sites identified as *PP "Gold" partners*. VOs get explicitly involved in the activity only in case of major updates. In case of minor updates the pilot activity is kept internal to the service infrastructure .

Pilot services for **minor updates** will be operated in production for 1 week. No artificial or focused "solicitation" of the service will be created. Only standard production activity + monitoring. In this case, running the pilot will be exactly like running the production service. The only extra commitment requested to the supporting production site is the awareness that the service is still experimental, so a prompt reaction is requested in case of problems in order to roll-back.

For **major updates** a preliminary negotiation between VOs, deployment teams and sites is necessary in order to agree on terms and conditions of the pilot activity. The negotiation is chaired and followed-up by the preproduction coordination team. The different phases of this negotiations consist into:

- identify suitable candidate sites among the PP Gold partners (or volunteers)
- provide information about the new features to the VOs and letting them express their interest into participating to the pilot activity. This is done through several channels, e.g. announcement during WLCG/EGEE Operations meeting; broadcast to VO Managers; direct communication to Experiment Integration and Support team (EIS).
- restrict (eventually) the rose of candidates/options and call a meeting to kick-off the deployment activity, During this meeting an agreement has to be reached about the timeline of the site to set-up

the service (e.g. 1 week) and the VO to give feedback (e.g. 2 weeks). Eventually the VOs may ask to be able to identify and select the sites providing the pilot services via jdl.

Once the pilot is started reminders for feedback are regularly sent to the VOs by the preproduction coordination.

The feedback provided by the VOs is taken into account, followed-up and eventually summarised and propagated the release notes by the release team.

In case no VOs commit to be active on the pilot service, or no feedback is received within the agreed timeline, the pilot service is evaluated with the same success criteria in use for the minor updates, and the decision to go to production is made internally by the operations team. In that case the release notes mention explicitly that **no feedback from users** was received during the pilot service.

As part of the preproduction service, deployment test of the update over significant deployment scenarios are run in parallel and separately from the pilot activity.

In case of updates to the services needed to fix critical bugs in production or security vulnerabilities the aforesaid policies may be overruled by joint decision of the EMT and PPS coordination.

Non-backward-compatible server update

Definition

With *non-backward-compatible (NBC) server update* we indicate all updates not in the previous categories. e.g.

- not compatible/usable with existing clients   need for a coordinated client update
- changes in the schema of the databases in back-end
- roll-back impossible or only partially achievable

General policies

The definition of this kind of pilots is highly variable and dependent on the use cases. In some cases more than one site will have to be involved, e.g. providing client and server previews. These updates will be treated on case basis.

Sites willing to support this pilot activities (PP "Platinum" partners) should have the expertise and the preparation to handle possible perturbation to the production environment. Throughout the duration of the preproduction activity, these sites must assure

- almost exclusive focus on the pilot activity
- concentration in time
- effective interaction with coordination and users

The mechanism to start and operate a pilot as well as the communication among the teams involved and the VOs is the same protocol defined for major backward compatible service upgrades, with possible changes to timescales due to the peculiarity of these events

For example:

- developers may be called to participate to the pilot kick-off meetings , in order to give hints and advices to involved sites, VO representatives, release manager, preproduction manager

- a "post mortem" should be done upon success or within 4 weeks from the beginning. In the post mortem an assessment is done and a decision is made about the follow-up (including decision to prolongate the testing time). Eventually general guidelines for the deployment can be drafted.

Introduction of a new service

The deployment of brand new services falls back in the category of backward compatible server updates and is dealt with using the same general workflows. Details in the duration and goals of the pilots may vary according to the complexity of the service to be deployed and have to be decided on case basis.

This topic: LCG > PreProductionUseCases
Topic revision: r17 - 2008-06-10 - AntonioRetico