Table of Contents

PPS brainstorming sessions

In this page we track meetings where general decisions about the gLite pre-production service are done

Mar 08: PPS re-structuring series

General goal: draw the guidelines for the re-organisation of the pre-production service in the aim to:

- Allow the VOs to use the PPS for testing
- Make a more effective usage of resources

Two sessions per week.

Written record of arguments used to make decisions

14-Mar-08 - Use case1: backward compatible client update.

Where: CERN

Who: Markus, Nick, Steve, Louis, Antonio

Preliminary assessment on resources

Resources currently in PPS (EGEE2): 13.6 FTEs over 25 sites (20 active in gridmap) + 1 FTE for coordination * Resources available for EGEE3 estimated in 9FTEs (~65% of quota in EGEE2)

Resources currently in distributed certification testbed (outside CERN): 4FTEs + 0.5 FTE for coordination * Resources available for EGEE3 estimated in 6-7FTEs (but they will work also on patch certification, not only on testbed adminitsration)

The SA3 distributed testbed will be re-scaled to two stable physical sites, 1 "virtual" site (resources at CERN) for quick patch certification and 1 testbed for stress testing (at CERN).

The distributed testbed should provide one CE for each batch system.

Distribution of backward compatible clients

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. This is more restrictive than necessary (the addition of new variables can be in most cases considered as backward compatible) . Considerations behind this restriction:

- updates where no configuration is needed for the clients are the most frequent cases
- they are also the cases where time-to-production is more important
- the added value of pre-deployment testing in PPS is the test of the release notes, but in a client update with no config instructions the release notes can be reduced to the minimum allowing, eventually, to skip the pre-deployment for these particular updates.

Therefore we decided to move the case of updates where *any* configuration changes are needed in the scenario2 "distribution of non-backward-compatible clients"

We want new clients to be distributed before certification to "willing" sites in production with the A1 mechanism.

The versioning of the installed clients on the site is handled by the script, but the versions are in general not synchronised with the status of the patch in certification. The "preview" installations use the same local environment

The distribution can be done on behalf of any VOs. OPS would work for sure, but as there is no unanimous consensus in the sites about this distribution mechanism, using ops (mandatory at all sites) may be controversial. One option could be dteam, where the privileged lcgadmin role has been cleaned-up recently. Another option could be to let the distribution be done by the VOs themselves

A mechanism to make the site publish a particular tag to indicate the available client versions is not in place

The patch handler (SA3 person) decides when a patch in certification (Status="In certification") can be "moved" to this deployment area. As a preconditions to be moved, the software must have passed a first round of basic tests although the full certification process my be not completed yet. The patches that have reached this level of maturity in certification can be flagged with a particular attribute in Savannah (e.g. preview=[Y|N]). The creation of a new state in the certification process (e.g. Status="Preview") is instead not advisable, because this would affect the workflow for all the services, also those ones for which the concept is not applicable.

Ideally, moving a client patch to attribute=Y should trigger the following actions

1. Creation of the tarball
2. Distribution to the sites (ideally using a SAM job) - as LHCb is doing
3. local testing (ideally using the same SAM job) - as LHCb is doing
4. publication of the tag in the information system (ideally using the same SAM job) - as LHCb is doing
5. update of a "release bulletin" documenting the versions available at the various sites (this can be done automatically on the PPS web site based on th e information extracted from Savannah)
6. notification to interested subjects (e.g. VOs)

Open question: what level of feedback are we allowed to expect/request from the VOs after the deployment? is there a way to formalise this interaction? P.S. If the patch is originated from a bug, in theory there is a way to backtrack the original in Savannah. In this case a message "ad personam" could be sent inviting the submitter to verify the provided solution.

After the certification, assuming that the release notes don't carry-on configuration information and that an installation test is carried out at an earlier stage than PPS, it is in theory possible to skip the pre-deployment test. The release notes have however to be checked before going to production. This effort is estimated in about 16 hours per year. There is a proposal to delegate this validation activity to the ROCs on account of the fact that they are formally responsible for the operations is pre-production. An exception was raised about the additional management effort needed to handle this rota (and sanction negligences) that could even outweigh the activity itself.

20-Mar-08 - Use case2: non-backward compatible client update and server update.

Where: CERN

Who: Markus, Nick, Louis, Antonio

Non-backward compatible client update

Definition: client updates where interventions on the environment or extra configuration is needed and so activity in required in some measure to the sites. When a client is not compatible with the existing service (and therefore a service upgrade is also needed) we include the case in the next category "non backward compatible server upgrade" In other words the new clients cannot be simply "pushed" to the sites by the deployment team. We will work with a reduced number of "friendly sites" in production willing to cooperate (the proposed list, based on the size of the size, the potential local interest and the "attitude" to experimentation is : CERN LIP RAL/MANCHESTER CNAF). The deployment at those sites will happen, as in the previous case, *before* the integration with YAIM. In most cases, however, the only intervention needed to the "friendly" site administrator will be to make a copy of the production setenv script in the appropriate path

---

NOTE: the pseudo code presumably requested to the user to use a client in prvew (e.g. identified by working version 126) should be

- select site supporting *126* from Infosys (to be confirmed)
- set path in submission client
- setenv

So "hooks" for the publishing in the infosys have to be created in the tarball releases

The deployment in the pilot of a new release will be announced by an e-mail. Separate acknowledgments by the collaborating sites are expected and an escalation mechanism (direct calls) has to be set-up for missing acks.

Once the configuration is confirmed to be done, we reconnect to point 5 in the previous case (same notification and feedback mechanism)

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 Upgrade

Definition: All updates that are compatible with the existing clients and don't introduce new functionalities. No changes in database schema. Updates can in general be rolled back. We distinguish between *minor* and *major* service updates

Minor: no new configuration parameters anywhere (not only in YAIM) AND < 5 bugs fixed AND (< equivalent of two days of programming - this needs an attribute to be defined, probably to be set by the patch manager)

Major: when any of the above conditions is false.

In order to set-up in general both sites an VOs need to get involved and go through a negotiation phase.

For the services the certification is a precondition. No pilots will be run on uncertified services.

Policy: We set up pilot services in both cases, but we get the VO explicitly involved in the functional testing only in case of major updates. In case of minor updates the pilot activity is kept internal. 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 the running the pilot will be exactly like running the production service. The only extra commitment requested to the supporting preproduction site is the awareness that the service is still experimental, so a prompt reaction in case of problems in order to roll-back.

For major updates a preliminary negotiation between VOs and sites is necessary to agree on terms and conditions of the VO activity. This activity will be coordinated by the release/PPS team. The goal of the negotiation is to set, in particular, a timeline for the VO to give feedback (e.g. 2 weeks). VOs must be put in the condition to select the sites providing the pilot services via jdl. Reminders for feedback to be sent during this time. If no feedback is received we fall back to the case of minor update and the decision to go to production is taken internally.

Non-backward compatible Server Upgrade

Definition: all updates not in the previous categories. e.g.

- not compatible/usable with existing clients ' ' need for a coordinated client update
- data schema changes
- roll-back impossible or partial

Deployment of brand new services falls in this category, although, with respect to the update, it is safer for participating sites: a failure of a new service has no impact on the production whereas a failure in and upgrade is potentially disruptive.

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 will be treated on case basis.

We define the guidelines of the communication process behind the start-up and operation of these pilots.

Sites willing to support this pilot activities have to be granted extra credits for the risk. Throughout the duration of the preproduction activity,they must assure, in the true "spirit of a pilot"

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

When the need for such an activity arises

- a pre-announce is done in order to gather possible volunteers among production sites and VOs
- a kick-off meeting is done with involved sites, VOs Release manager, preproduction manager (and developers) in order to agree on terms and conditions of the activity
- post mortem after success or within 4 weeks. An assessment is done and a decision is made about the follow-up (including decision to prolongate the testing time) eventually guidelines for the deployment are drafted

Implementation notes

Credit-based accounting for sites participating to the PPS.

"Idle" PPS sites to be reconverted to "potential" pilot sites (pre-production Club). Human resources at these sites normally working in production but ready to be shifted from production to preproduction if the site gets involved in a pilot (production and preproduction resources are not neatly distinguished anymore in EGEE3) Registration in GOCDB no more necessary for PPS sites ("technical" registration to pps to be replaced somehow)

Statements and decisions

1. We will use the A1 mechanism to distribute clients in some collaborating production sites
    - ♦ This is done because we observed that on mission-critical client updates, the real usage of the new clients by the VOs was done in the "shared area" and not in preproduction, which, in

fact, was always far behind

2. A subject in SA3 (the patch handler) will be responsible to decide when a version of the client can go in "preview"
    - this cannot be done immediately after delivery of the patch to certification because too unstable, nor after the delivery to preproduction, because too late
3. "pilots" and deployment tests run in parallel because they are focused on different aspects (functionality and installation)

Re-modeling of pre-deployment reports

14-Mar-08 - first draft of a data model for deployment testing.

Where: EVO

Who: Mario, Esteban, Alvaro, Louis, Antonio

1.) we want to use a database as back-end for the results of the pre-deployment tests because we reckon that the db on wiki is not going to scale
2.) based on the conceptual model implemented in the wiki page we drafted a data model for the pre-deployment test databases
3.) some of the pre-defined information in this database (namely the population of service types, service and OS versions etc.) can and should be extracted directly from Savannah via an inerface to the Savannah db dumb
4.) the php pages will be integrated in the context of the PPS website, which is already based on a php script. That should help sorting out possible integrations issues
5.) the physical database will be for the time being located at CESGA
6.) CESGA, with the help of Pablo who implemented the accounting system interface, starts working on a prototype of the database and the interfaces (probably some php pages on top of it). 15th of April is the proposed deadline for the prototype db to be ready.
7.) Antonio will be informed of the progresses and will stay in touch available for questions and explanation on the system. He will also be the link with the PPS re-organization working group by forwarding info about analysis work and possible new requirements being drawn in that context.
8.) As soon as the first instance of the database will be available, Antonio will start working to its connection with the Savannah dump to populate the menus

---

This topic: LCG > PPSBrainstormingSessions
Topic revision: r6 - 2009-05-30 - AntonioRetico