

Deployment strategy of Beamline and Experiment Control (BEC) components across development and production environments

Ivan Usov¹, Borys Sharapov², Klaus Wakonig³, Leonardo Sala², Simon Ebner⁴

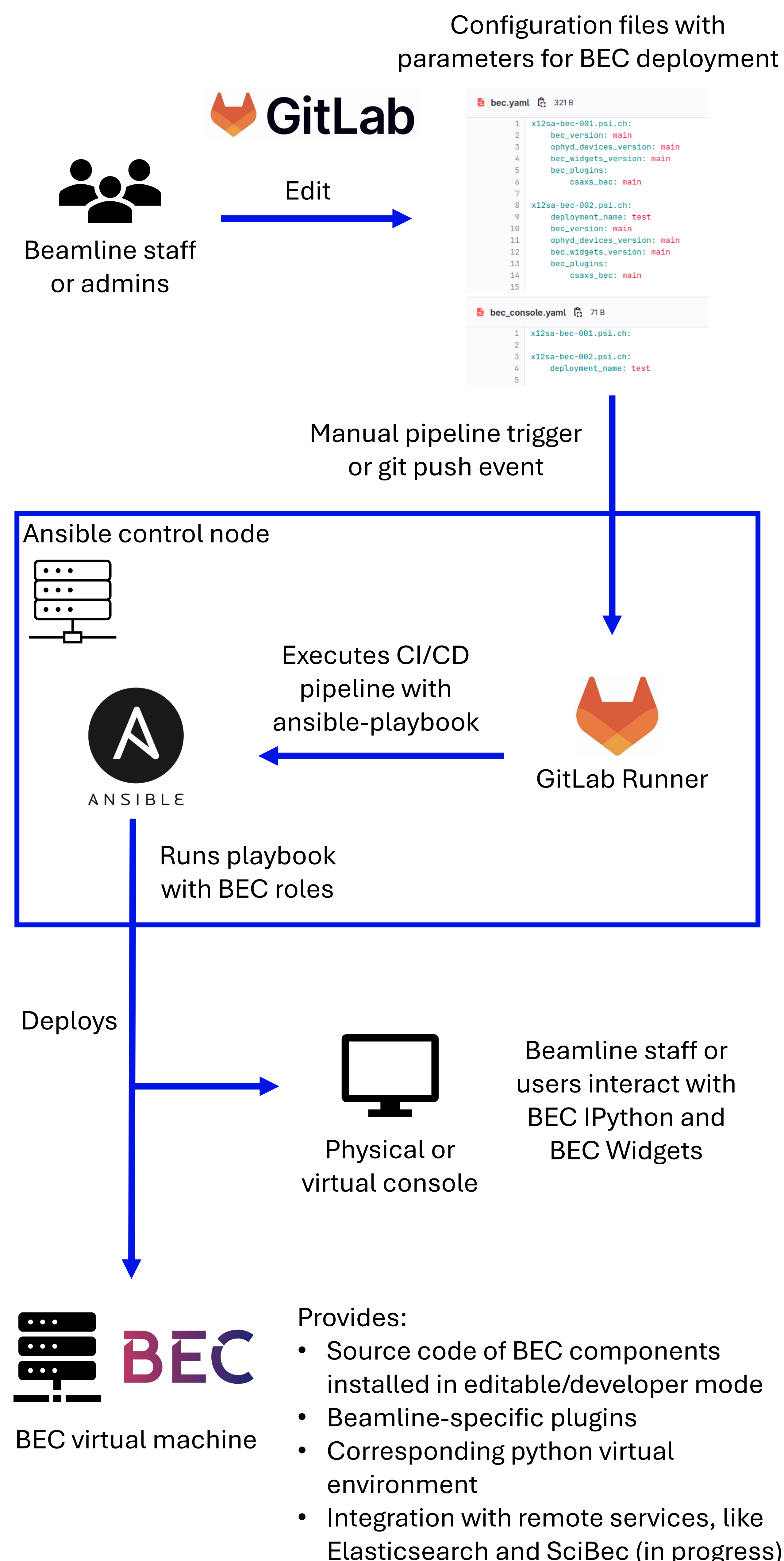
¹Data Processing Development and Consulting, Science IT Infrastructure and Services, Paul Scherrer Institute

²Data Analysis and Research Infrastructure, Science IT Infrastructure and Services, Paul Scherrer Institute

³Experiment IT Development and Operations, Science IT Infrastructure and Services, Paul Scherrer Institute

⁴Core Linux Research Services, Science IT Infrastructure and Services, Paul Scherrer Institute

We present a deployment strategy for BEC components and dependencies, leveraging on-premise GitLab pipelines, runners, and Ansible roles/playbooks. Combining GitLab's continuous integration/continuous deployment (CI/CD) automation with Ansible capabilities, we expect Beamline and Experiment Control (BEC) ecosystem to achieve a scalable deployment mechanism across all SLS 2.0 beamlines, facilitating adaptation to evolving requirements and ensuring optimal user configuration interface.



GitLab

- On-premise internal GitLab instance <https://git.psi.ch> with access to local GitLab runners
- Beamline managers with write access (via GitLab Web IDE or local git repositories) to their respective git repository with BEC deployment configuration files
- GitLab Pipelines are read-only through a reconfigured “CI/CD configuration file” pointing to an external repository

BEC deployment configuration

- A simple user interface with declarative yaml files
- The configuration is defined on a per-host basis and specifies versions of BEC components and beamline plugins to be installed in each deployment

GitLab Runner and CI/CD pipelines

- GitLab Runner is installed and configured on an ansible control node
- Can be triggered by a git push event or from GitLab web interface
- Pipeline customization for manual pipeline runs, e.g., to limit execution to a list of hostnames for only a specific service redeployment, and/or partial service redeployment
- GitLab Runner executes a CI/CD pipeline that, in-turn, runs an Ansible playbook
- The setup is similar between the development and production environments, differing only in the Ansible control nodes and the set of defined hosts

Ansible

- Execution of a playbook with imported *psi.bec* and *psi.bec_console* ansible roles
- An access to defined BEC servers and consoles is secured via ssh keys
- A straightforward scaling to other service deployments by including additional ansible roles in the playbook of a specific beamline

Virtual machines

- Procured with VMWare and configured with Puppet
- Run Red Hat Enterprise Linux 8 (RHEL8)
- Firewalled within a beamline subnetwork
- Monitoring with Icinga2

BEC

The Beamline and Experiment Control (BEC) is a new python-based control system for experiments that targets the Swiss Light Source upgrade (SLS 2.0) at Paul Scherrer Institute.

- <https://bec.readthedocs.io/en/latest/>