# Windows Terminal Servers Orchestration

Sebastian Bukowiec, Ricardo Gaspar, Tim Smith (CERN, IT-CDA)

**Project objective:**

automate and orchestrate common activities by defining workflows to reduce error-prone manual actions and operational workload

## Microsoft® Orchestrator

Orchestrator is intended for automation of all on-premises resources. It uses a different runbook engine than Service Management Automation (SMA). Orchestrator relies on a graphical interface to create runbooks. Lack of scripting is one of the reasons to build CERN Windows automation around SMA. Orchestrator is a mature product with many Integration Packs ready to be used.

**Infrastructure:**

- 28 load balanced clusters
- 106 Windows Terminal Servers
- 2 highly available load balancers

## Microsoft® Configuration Manager

Configuration Manager provides a comprehensive solution for change and configuration management. Configuration Manager allows to deploy operating systems, software applications, and software updates. Although not currently implemented, Configuration Manager will be integrated in the workflow in the near future.

## Microsoft® Operation Manager

Operations Manager provides Windows infrastructure monitoring that is flexible, helps ensure the predictable performance and availability of vital applications, and offers comprehensive monitoring for datacenter. SCOM can be extended by importing management packs (MPs) which define how SCOM monitors systems.

## Microsoft® Service Management Automation with Windows Azure Portal

openstack™ CentOS HAPROXY

Highly available and scalable infrastructure engine

Encrypted credentials stored centrally in the database

Integration into existing systems

Workflows can run on schedule

powered by **PowerShell workflows**

GitLab

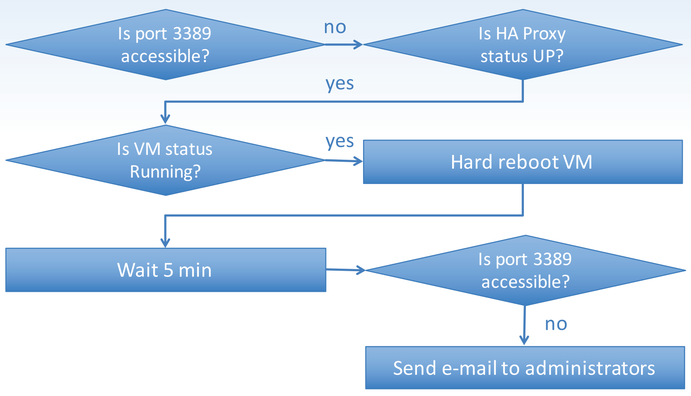
Git PUSH → GitLab Continuous Integration (CI) with Windows Runner

PowerShell using SMA module to sync code

## Simplified Orchestration | Workflow Examples

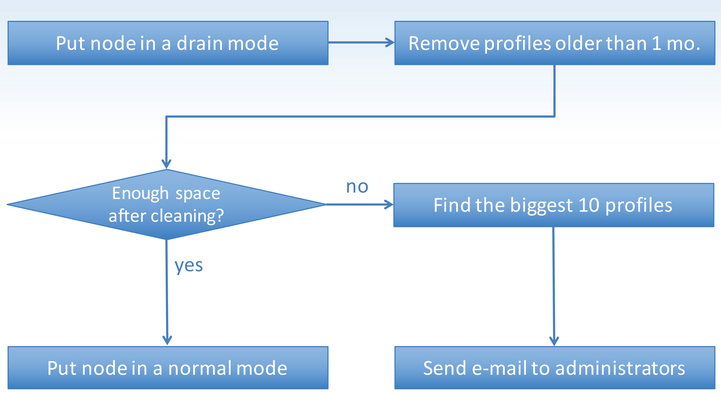
### Use case

node replies to ping but no connectivity on RDP and no GNI alarm



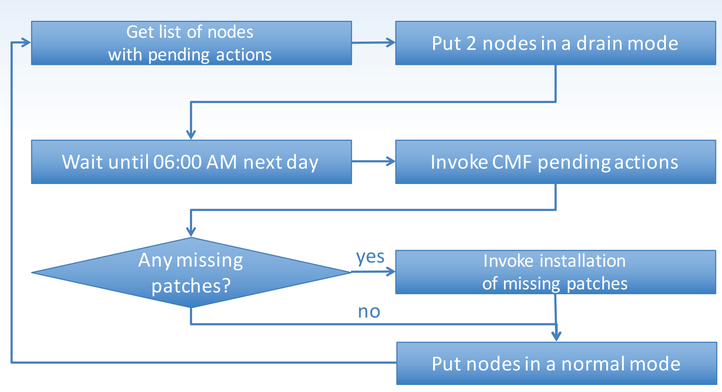

### Use case

Operation Manager alarm: low space on D: drive (user profiles store, ~600 profiles per node)




### Use case

installation of general purpose cluster patches, updates and applications




HAPROXY

To provide **redundancy and scalability** Windows Terminal Servers are deployed in form of **clusters**. Each cluster contains a minimum of two nodes. For all clusters the connections are **load balanced** by two HA Proxy servers (**active-active**). To provide highly available and **resilient infrastructure** servers utilize **keepalived** software. Each node reports its load to HA Proxy via **feedback agent**. When the load reaches 90% the node is automatically put in drain mode.

For **compliance** with policies and Windows Terminal Servers (RDS) settings each machine is member of a **Puppet-controlled infrastructure**. The baseline configuration is **applied per host group**. Each cluster has its own dedicated host group. Host group management is done via Foreman software. The baseline is in the form of Puppet code that is version controlled and stored in a **Git repository** (CERN GitLab service).

puppet

openstack™

Most of Windows Terminal Servers are deployed as **virtual machines** in the CERN Cloud Infrastructure **powered by OpenStack** software. They are hosted in projects assigned to **different locations** in CERN computer center for **higher availability**. Failure of one hypervisor or network switch will not affect cluster availability. Windows Terminal Servers are deployed with **standard images** customized via Puppet.