Integrated automation for configuration management and operations in the ATLAS online computing farm

A. Amirkhanov[1] · S. Ballestrero[2] · F. Brasolin[3] · C. J. Lee[4] · H. du Plessis[5] · K. Mitrogeorgos[6] · M. Pernigotti[7] · A. Sanchez Pineda[8] · D. A. Scannicchio[9] · M. S. Twomey[10]

1. Budker Institute of Nuclear Physics, Novosibirsk · 2. University of Johannesburg · 3. INFN Sezione di Bologna · 4. University of Cape Town · 5. University of Johannesburg · 6. Aristotle University of Thessaloniki · 7. CERN · 8. INFN Gruppo Collegato di Udine · 9. University of California, Irvine · 10. University of Washington

Why

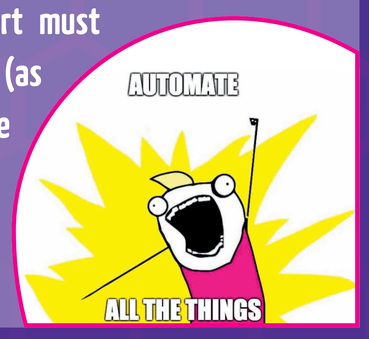
The online farm of the ATLAS experiment at the LHC consists of nearly 4000 servers with various characteristics.

Due to the large scale of the farm and the variety of the systems, appropriate tools to address various requirements are needed to effectively manage and monitor these servers. As a result, when experts are performing routine interventions, they are required to update a number of disparate tools (removing from production, scheduling downtime, etc).

Currently, some of these tools are:

- ConfDB - in-house configuration management tool
- Icinga2 - monitoring and health reporting system
- OKS (Object Kernel Support) - an object-based in-house database defining the resources available for ATLAS data taking

This is a time consuming process, and the expert must remember to update all the tools in the correct order (as per the defined procedures). Additionally, the procedures require the expert to constantly "check up" on the server to determine when it is ready for an intervention, and hence results in wasted time.



How

To ensure the longevity of the planned system, a community-developed and supported tool was preferred (compared to a custom-developed in-house tool). After evaluating various solutions on the market, Rundeck, a tool which automates jobs, was selected for implementation and further evaluation.

A number of scripts and plugins needed to be written to enable Rundeck to interact with the various tools. These include:

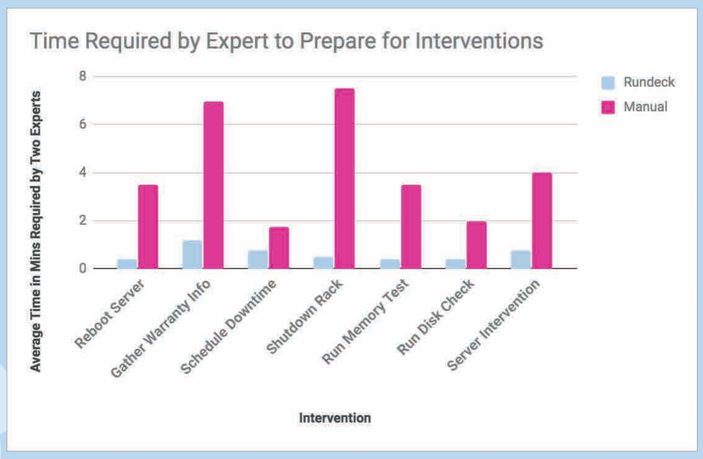
- **Node source script** - regularly updates the list of servers that Rundeck can operate on with live production status
- **ConfDB plugin** - uses the ConfDB API to read and modify the server's current production status
- **Icinga2 plugin** - uses the Icinga2 API to schedule and remove downtimes, as well as run health checks for a server
- **OKS plugin** - interacts with the OKS database to request that a server be inserted or removed from data processing
- **Notify plugin** - triggers Webhooks (arbitrary HTTP callbacks) and sends SMS, email, and Redmine (ticketing system) notifications
- **Waitfor plugin** - allows a Rundeck job to be paused until a server is in a desired state or until a manual intervention has been completed

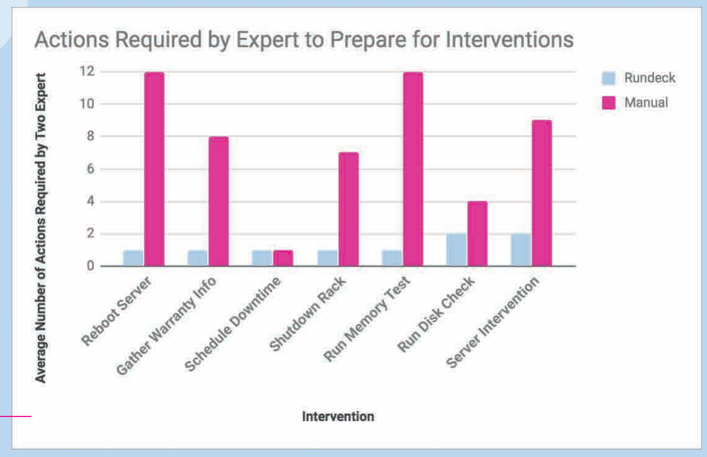
This was tested first as a small pilot project in TestBed, a 400 servers lab environment. Finally, the tool was deployed in mid-2018 within ATLAS where it is able to automate operations across 4000 servers.

What Now

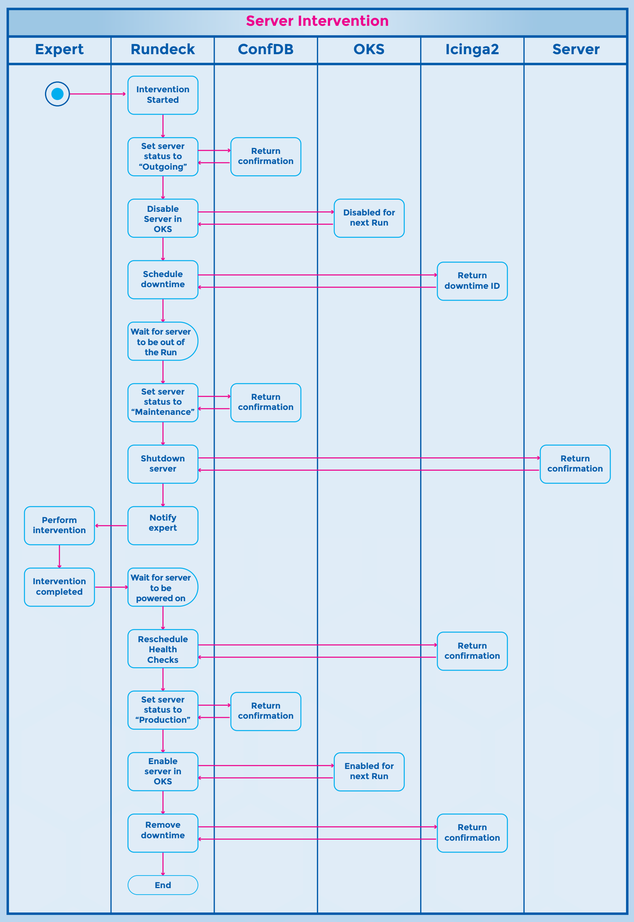
Rundeck has been deployed in both the test and production environments. A number of jobs have been created in order to simplify daily operations of the team. Most operations have been reduced from many steps to simply launching the task from Rundeck. Additionally, the time taken by an expert has been reduced, as Rundeck is able to follow the execution of the intervention, leaving the expert free to continue with other tasks.

By using a shared platform for all interventions, experts are also able to see the status of interventions that were initiated by other colleagues, thereby preventing work from being duplicated.



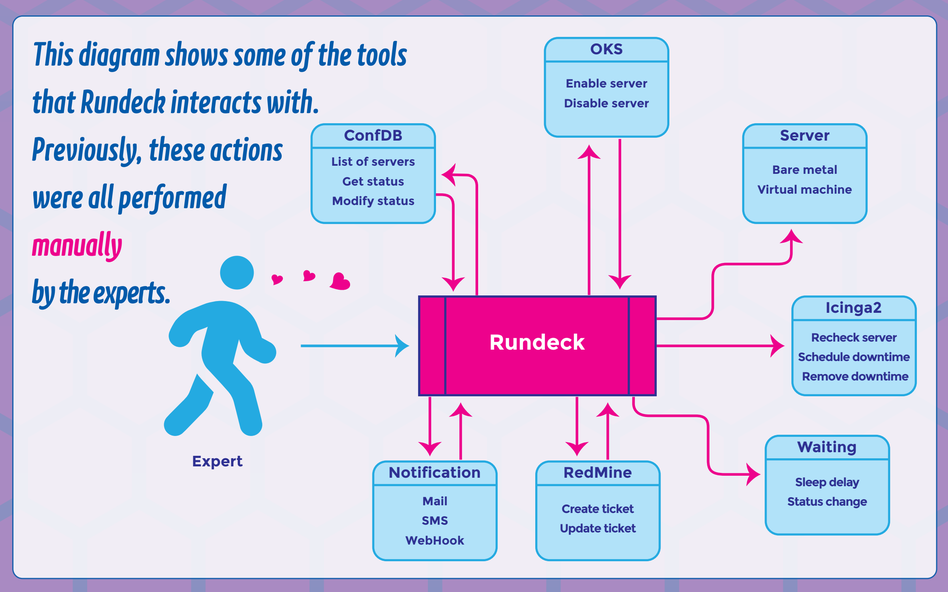


Differences in time (plot above) and number of actions required (plot below) when performing a number of common interventions.



A typical workflow of actions performed while intervening on a server.

System Interactions



Design: Nadia Neri

Conclusions

Implementing an automation tool, like Rundeck, has reduced the number of manual processes that an expert must perform when carrying out an intervention. As a result, the expert's workflow would not be interrupted to regularly check the status of each step before proceeding with the next one.

Although some plugins had to be developed to interact with various systems, this was easily accomplished through Rundeck's plugin framework. Unfortunately, some effort was needed to work around Rundeck's shortcomings (such as not being able to pause a job). However, Rundeck has proven itself to be a valuable tool within the team.

Since its implementation, Rundeck reduced the risk of human error by ensuring that correct procedures are followed for each intervention, whilst maintaining a clear audit trail of every action taken. With Rundeck handling many aspects of common interventions, experts are interrupted less, thereby allowing them to focus more on other tasks.