EPJ Web of Conferences **295**, 07038 (2024) https://doi.org/10.1051/epjconf/202429507038
CHEP 2023

Preparing for a new Data Center: Automated Management of a 10'000 node bare-metal fleet in CERN IT

Maryna Savchenko[1,*], *Luca* Atzori[1,**], *Nikos* Papakyprianou[1,***], *Michał* Piszczek[1,****], and *Arne* Wiebalck[1,†]

[1]IT Department, CERN - 1211 Geneva 23 - Switzerland

Abstract. CERN IT has consolidated all life-cycle management of its physical server fleet on the Ironic bare-metal API. From the initial registration upon the first boot, over the inventory checking, the burn-in and the benchmarking for acceptance, the provisioning to the end users and the repairs during its service, up to the retirement at the end of the servers' life, all stages can be managed within this framework. This article will walk you through the server's life in the CERN data center, and explain how this enables us to handle a fleet of almost 10,000 nodes in an automated and efficient way and to prepare for the new data center which is currently being built. Eventually, we will round up things with the top challenges we faced when moving to this system, like the transparent adoption of already in-production nodes, after-the-fact inventory updates, or scaling issues.

1 Introduction

Almost 10 years ago CERN IT successfully moved to an on-premise cloud, based on OpenStack. The concomitant policy introduced back then was to have all resources provisioned as virtual machines. While this worked out for almost all use cases, there was always a need to provision bare-metal servers in addition [1]: for some services, it was not economical to use virtual machines (e.g. when full node Virtual Machines (VMs) were required), it did not make sense technically (e.g. for storage servers), it introduced a potentially undesired dependency or lack of control (e.g. core networking services), or it was simply not feasible (e.g. cloud hypervisors). Equally, some use cases could not be fulfilled by virtual machines, e.g. evaluating fine-grained performance tuning which would be disturbed by crosstalk from co-hosted instances. In some cases, like the batch compute service, the choice of virtual vs. physical resources was far from obvious (the gain of having an API to manage resources in an automated way outweighed eventually the performance loss due to the virtualization tax), and could have gone the other way as well. As a consequence, there was always a need for bare-metal provisioning workflows. OpenStack was already used for managing virtual machines in the Cloud, so the obvious solution was integrating Ironic which is an OpenStack project for provisioning bare-metal (as opposed to virtual) machines [2].

*e-mail: maryna.savchenko@cern.ch
**e-mail: luca.atzori@cern.ch
***e-mail: nikolaos.papakyprianou@cern.ch
****e-mail: piszczke@gmail.com
†e-mail: arne.wiebalck@cern.ch

Currently, all the new deliveries are managed by Ironic making it around 9000 physical servers, among which there are around 1700 hypervisors hosting around 13500 VMs. Ironic allows to manage such amount of nodes with minimal human interaction, almost the whole life-cycle of the servers is managed automatically by Ironic. By the end of 2023, the new computing center in Prevessin is planned to receive new servers and the Cloud team is configuring another OpenStack region there.

2 Ironic overview

Ironic is an open-source project that fully manages bare-metal infrastructure. One of its benefits is being API-centric. It includes a complete set of RESTful APIs, providing a standard interface regardless of the vendor. Ironic allows the provisioning and management of bare-metal machines. It discovers bare-metal nodes, catalogs them in a management database, and manages the entire server lifecycle including enrolling, provisioning, maintenance and decommissioning.

Ironic can be used standalone or integrated into an OpenStack deployment. As CERN IT was already providing virtual resources via OpenStack, the decision to integrate was taken. From a user point of view, it allows interaction with physical machines the same way as with virtual machines. Moreover, the exact same workflow for requesting and applying virtual resources can be now used for physical ones which provides a better user experience.

Figure 1 shows one of the common use cases - a workflow of a user requesting physical resources from Ironic when it is integrated into Nova, the OpenStack Compute project.

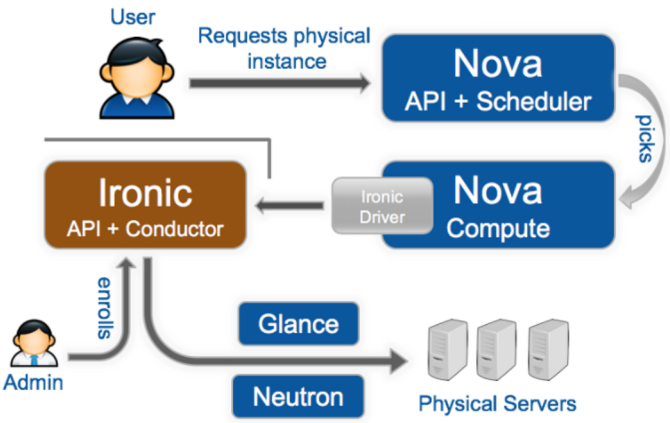

Figure 1. Requesting physical resources through OpenStack [3]

3 Server Life-cycle Management

The server life cycle at CERN consists of several parts. The first steps are a specification meeting with a discussion of the hardware characteristics and an invitation to tender and contract awards. With the delivery, this process usually takes around 9 months. After this, there is a physical installation and an acceptance period where all the servers are tested which usually takes around 5 weeks. Finally, as shown in Figure 2, the servers are being used by the users and after 5 years they are being cleaned and removed from OpenStack and databases as well as physically from the data center.

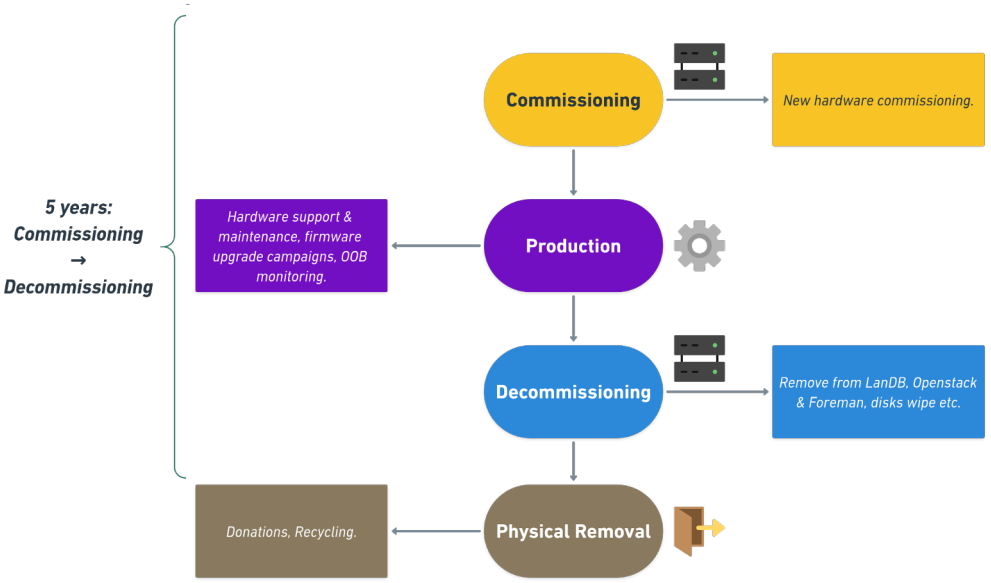

Figure 2. Commissioning to decommissioning workflow

The goal when introducing Ironic was to automate as many stages as possible. Some of these elements have to be done manually like the discussion of the hardware specifications, comparing vendors, etc. But from the moment of the physical installation of the servers, Ironic takes over.

Figure 3 shows which parts of the server's life are taken care of by Ironic.

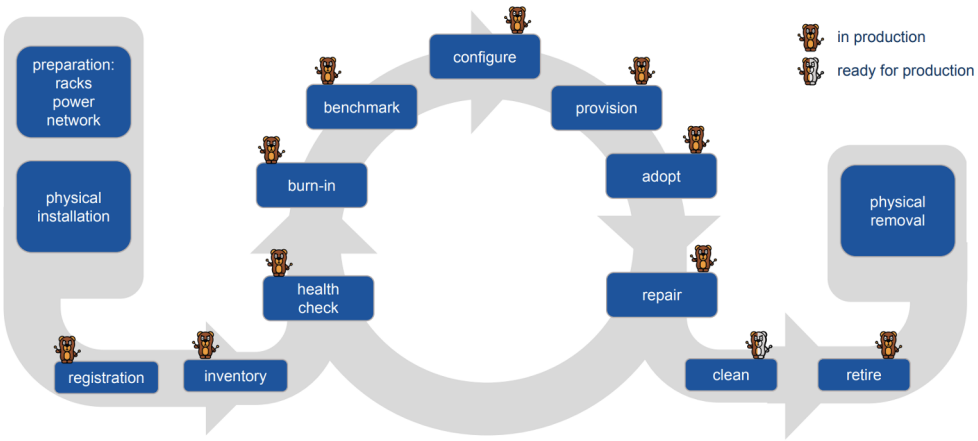

Figure 3. Server Life-cycle Management

Almost the whole life-cycle of servers is covered by Ironic. As soon as they boot up, they will be automatically enrolled into Ironic and a CERN database, next benchmarked and stress tested. After the acceptance period, they are provided to the end user where they can be used until the end of the warranty. In case of failures Ironic simplifies the process of debugging

and repair as well as it allows the Cloud team to easily configure nodes, reinstall, or change the ownership, and more.

While Ironic comes with support for server retirements, this is not fully deployed yet, and the cleaning of the disks before physical removal is done via a separate process. The future plan is to include it in the Ironic workflow which would make the whole server life managed by Ironic.

3.1 Auto-Registration with Ironic

As soon as servers are installed and powered on in the data center, they are automatically enrolled in Ironic. To achieve this, unregistered nodes boot using PXE (Preboot Execution Environment) into the IPA (Ironic Python Agent) image pointed by DHCP as soon as the nodes send a 'discover' packet.

After this, Ironic will register servers in the CERN database called LanDB, run the introspection which will collect the data about the hardware and send it to the storage and finally enroll them into Ironic.

A more detailed workflow of a server auto-registration is presented in Figure 4.

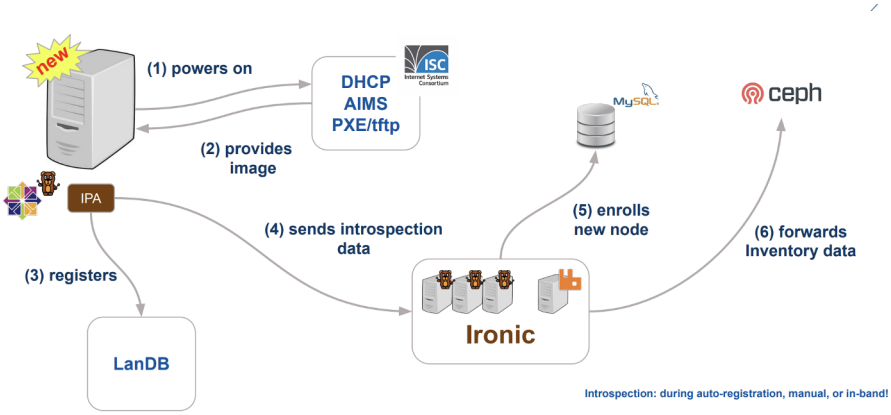

Figure 4. Server auto-registration workflow

3.2 Hardware Inventory

One of the services Ironic provides is introspection, a tool for discovering hardware properties for a node managed by Ironic. The data backend in the deployment is S3-based and is integrated with OpenDCIM (Data Center Inventory Management application) which can be later retrieved through the CLI or the webUI. To run it, a node has to boot into the IPA image.

This will not work in the case of the situation when the initial introspection did not collect the data correctly and it got noticed only when the server was already instantiated by a user and is being used. This was one of the challenges the Cloud team faced in the deployment when the wrong serial number or a bug in the *lshw* hardware reporting tool (reporting wrong memory) led to missing inventory data. The solution was introducing active introspection [4] which allows running the Ironic introspection on the nodes in an ACTIVE (i.e. instantiated by a user) state. To do this, one has to login into the node, install all the needed packages and run the introspection. An alternative solution is creating a container with all the required packages so there is no need to install them separately every time.

3.3 Hardware burn-in

After the nodes are enrolled into Ironic, the burn-in is executed to check their capabilities and to try to provoke early failure via stress tests. This part was integrated into Ironic as customized cleaning steps. This includes CPU and memory burn-in using *stress-ng* [5] tool and disk, network burn-in using *fio* [6]. For the real-time log handling the Cloud team is using *Fluentd*, *Elasticsearch* and *Kibana*. All of them are configurable, meaning that different parameters can be modified using the "driver-info" field.

3.3.1 Network burn-in

For the network burn-in, things are more complicated because it requires pairing between writer and reader nodes. There are 2 options: the initial version uses static pairs - each pair has to be set statically and if, for example, one server from the pair is not responsive, the second one will not have the possibility to be tested. The second option is dynamic pairing with Zookeeper, which provides a distributed coordination backend. This allows burn-in to proceed on a "first come first served" basis with the nodes available, rather than a node being blocked since the static partner is currently delayed.

3.4 Benchmarking

Another customized cleaning step is benchmarking. It downloads and executes a bash script which runs a selected benchmark and uploads the results to Open Search. In the CERN Cloud case, this is a benchmark specific to the High-Energy-Physics [7] use cases that are designed to scale with the performances of the high-energy physics codes on similar machines.

4 Ironic Evolution over Time at CERN

Figure 5 shows the number of Ironic nodes through time in the CERN deployment over time. Arrows mark different reasons for their growth or shrinking like new deliveries, an adoption campaign, retirements and container switch-off for saving energy purposes.

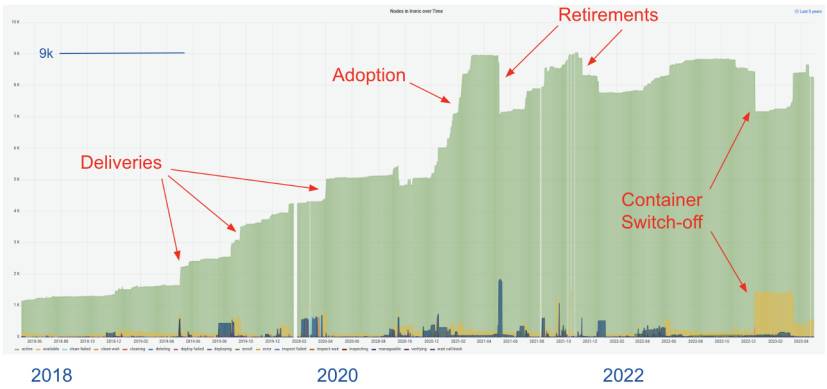

Figure 5. Number of Ironic managed nodes

4.1 Adoption of in-production physical nodes into Ironic and Nova

It is not uncommon that operators set up Ironic already having some physical nodes managed by other tools. The CERN Cloud was not an exception but as there was a plan of managing all the physical nodes by Ironic, it was needed to enroll already existing nodes. It was impossible to use a standard enrolling path as it included complete cleaning of nodes. And while Ironic provides a possibility of adoption on its own, Nova does not.

The basic idea to solve this was to create and correlate Nova instances with the correct Ironic physical nodes without touching the underlying nodes in Ironic. To achieve this, the *fake-hardware* type and the *fake drivers* were used.

The fundamental steps in the procedure were [8]:

- create a bare-metal flavor (specification defining a hardware configuration)
- create a hosting project
- enroll the nodes into Ironic
- change the hardware type and the interfaces to fake drivers which prevents nodes from being physically cleaned
- one by one, add the nodes to the placement aggregate and create the instances
- change the hardware type and the interfaces back to the real ones

An important step during the instance creation in Nova was to match the physical servers' current names and schedule them on the matching machines. This was achieved by adding nodes to the placement aggregate one by one, leaving Nova with only one choice.

4.2 Scaling Ironic with Conductor Groups

Ironic was introduced in 2018 at CERN IT. With its growth, various scaling issues were hit. One of the main reasons for this was the time that the Resource Tracker inside the nova-compute service was taking to loop through all the resources [9]. Its job is to update the OpenStack Placement service with the current state of the resource providers, e.g. which nodes are currently available for instantiation. This is a blocking operation and when the Cloud deployment was counting around 5000 nodes it took almost 3 hours to complete, meaning that for 3 hours users were not able to create or delete instances or change their state.

The Nova/Ironic deployment architecture was partly responsible for this: all physical nodes were managed via a single "bare-metal" cell controlled by a single "nova-compute" node that interacts with Ironic. The instance creation time was reduced by splitting the deployment into conductor groups which were introduced in the "Stein" release. It allows splitting the infrastructure and having failure domains. With conductors groups, it became possible to map "nova-compute" nodes to a set of Ironic nodes.

Based on timing measurements, the infrastructure was split into groups of around 500 Ironic nodes each: for groups of this size, the "nova-compute" resource tracker cycle takes around 15 minutes to complete. This seemed a good compromise for the CERN-specific use case and user expectations.

4.3 Providing non-x86 resources

The majority of the resources in the CERN data centers are x86_64 based, but the demand for heterogeneous architectures (e.g. GPUs or other processor architectures such as ARM) has increased during the past years. While the first GPUs were deployed several years ago, the first ARM nodes were enrolled into Ironic during 2022.

4.3.1 ARM resources

Providing ARM servers in Ironic included challenges like building a cloud Linux image, an IPA image and an adaptation to the PXE boot service.

Building a cloud image included bootstrapping AArch64 as there was no physical ARM server at the moment of image building. To achieve this, several steps were required:

- a modified QEMU emulator was used to emulate AArch64 architecture on Intel
- building RPM packages with koji builder
- installing a VM using a kickstart file
- snapshot of the VM transformed into a cloud image for ARM

To support an ARM architecture, the workflow was changed to build two separate images for AArch64 and x86_64 instead of building a single IPA image. More details on the ARM integration can be found here [10].

5 Conclusion

Even though CERN IT successfully moved to an on-premise cloud, based on OpenStack, it was clear that there would be always a use case for provisioning bare-metal servers. Initially, the choice to use Ironic was due to its strong integration with OpenStack, which had already been used at CERN for a couple of years. The amount of physical servers at CERN has been constantly increasing which caused Ironic to grow into a bare-metal management framework, enabling Infrastructure-as-Code tools, such as Terraform, on top, while handling the full life-cycle of physical servers in CERN IT, from initial registration to final retirement. Additionally, being a community-based software Ironic continuously extends its functionality to support more use cases. Overall, Ironic is an excellent choice for managing an extensive fleet of physical servers at CERN IT, and it is planned to be extended to the new computing center currently being built in the CERN site of Prevessin.

References

[1] A. Wiebalck, *The Case of Ironic in CERN IT*, https://techblog.web.cern.ch/techblog/post/why-ironic/, [Online; accessed 23-August-2023]

[2] *Ironic*, https://ironicbaremetal.org/, [Online; accessed 23-August-2023]

[3] A. Wiebalck, B. Moreira, S. Cai, *Scaling Bare Metal Provisioning with Nova and Ironic at CERN: Challenges and Solutions*, https://superuser.openinfra.dev/articles/scaling-bare-metal-provisioning-with-nova-and-ironic-at-cern-challenges-solutions/ (2021), [Online; accessed 23-August-2023]

[4] *Active introspection*, https://docs.openstack.org/ironic-python-agent/latest/admin/how_it_works.html, [Online; accessed 23-August-2023]

[5] *stress-ng*, https://wiki.ubuntu.com/Kernel/Reference/stress-ng, [Online; accessed 17-September-2023]

[6] A.C. Jens Axboe, *fio(1) - Linux man page*, https://linux.die.net/man/1/fio, [Online; accessed 17-September-2023]

[7] *How to Run HEP-SPEC06 Benchmark*, http://w3.hepix.org/benchmarking/how_to_run_hs06.html, [Online; accessed 18-September-2023]

[8] A. Wiebalck, *Adoption of in-production physical nodes into Ironic and Nova*, https://techblog.web.cern.ch/techblog/post/ironic-nova-adoption/, [Online; accessed 23-August-2023]

[9] A.W. Belmiro Moreira, *Scaling Ironic with Conductor Groups*, https://techblog.web.cern.ch/techblog/post/conductor-groups/, [Online; accessed 23-August-2023]

[10] M. Savchenko, *Providing ARM and GPU resources in the CERN Private Cloud Infrastructure*, https://indico.cern.ch/event/1222948/contributions/5321051/ (2023), [Online; accessed 23-August-2023]