ATS-IT: Kubernetes Phase 2 Architecture Overview

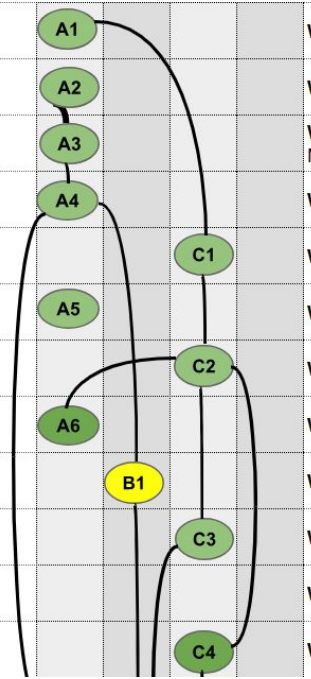
Journey's Timeline

First steps with containerisation
First steps with orchestration
Pilot services on Kubernetes

Road toward production
Controls Datacentre infrastructure refresh

Running of the first production Accelerator Controls Services

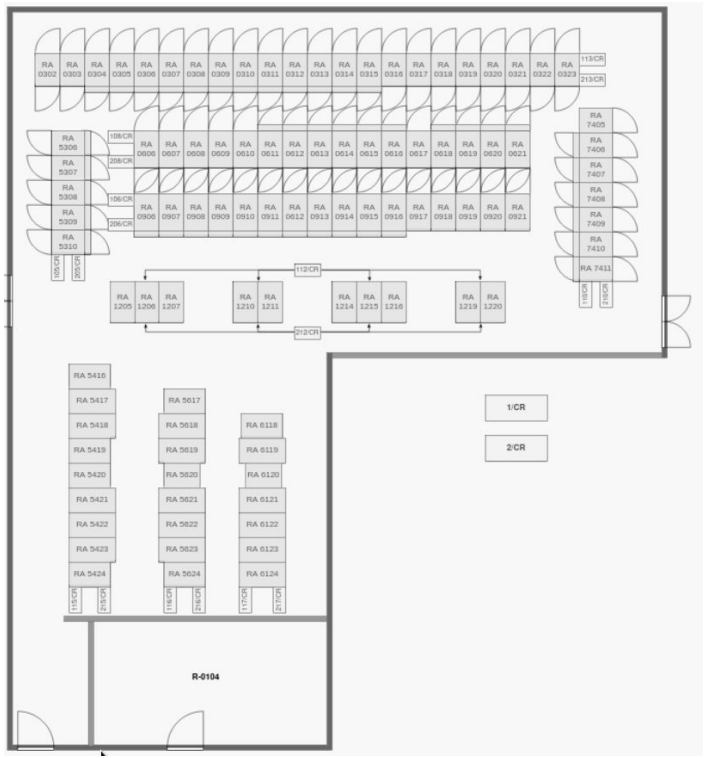
Transition of all remaining Accelerator Controls Services

2023-Sep-01	2023-Sep-01		**When the Phase 2 PSO has been approved and the responsibilities and stakeholders defined.**
2023-Oct-05	2023-Oct-05		**When the Phase 2 stakeholders have been onboarded.**
2023-Oct-05	2023-Oct-05		**When the responsibilities for the operational support and maintenance of the targeted platform have been defined.** Must ensure it is aligned with service levels for the existing services and infrastructure.
2023-Oct-05	2023-Oct-05		**When the long term resources required for support in IT have been identified and communicated to the ATS-IT SC.**
2023-Oct-30	2023-Oct-30		**When the remaining technical tasks from Phase 1 have been completed.**
2023-Nov-03	2023-Nov-03		**When the work from Phase 1 has been presented in IT (ITTF) and ATS forums.**
2023-Nov-15	2023-Nov-15		**When the control plane for the TN/CCR region has been merged into the standard service deployment.**
2023-Nov-15	2024-May-01		**When BE-CSS has been onboarded into service management, design and evolution.**
2023-Nov-30	2024-Apr-30		**When the foreseen architecture has been presented to all stakeholders.**
2023-Nov-30	2023-Nov-30		**When integration and functional tests have been enabled in the TN/CCR region.**
2023-Nov-30	2024-Mar-30		**When secret management solution has been chosen and integrated.**
2023-Dec-15	2024-Apr-30		**When detailed topology from OpenDCIM has been integrated.**

2023-Dec-15		A7	When the first BE-CSS systems targeted to run on Kubernetes have been defined and approved by all stakeholders.
2023-Dec-15	2024-Apr-30	A8	When Kubernetes-era DevSecOps for BE-CSS have been defined, presented and approved by stakeholders.
2023-Dec-15	2024-Mar-30	D1	When the image registry is declared production ready (inc. sync and replication, performance).
2024-Jan-30	2024-Jun-30	C6	When failures scenarios and service availability have been validated.
2024-Jan-31	2024-Jun-30	B2	When the future architecture has been approved by all stakeholders, including the IT Architecture Review Board (ARB).
2024-Feb-01	2024-May-15	C5	When integration with COSMOS has been completed.
2024-Feb-01	2024-Jun-01	D2	When secret management solution is production ready and end user documentation is available.
2024-Mar-01	2024-Jun-30	D3	When the previously identified BE-CSS controls systems are put in production.
	2024-Jun-30	D4	When the architecture and support structure have been endorsed by the ATS-IT SC.

Controls Data Centre

- The Controls Datacenter hosts critical Accelerator services
  - 25 racks
  - 500 bare-metal servers
  - **Isolated network (Technical Network)**
- All hardware is procured with IT
- Major upgrades constrained to accelerator schedules
- Diverse in-house DevSecOps practices
- Underused CPU and memory resources



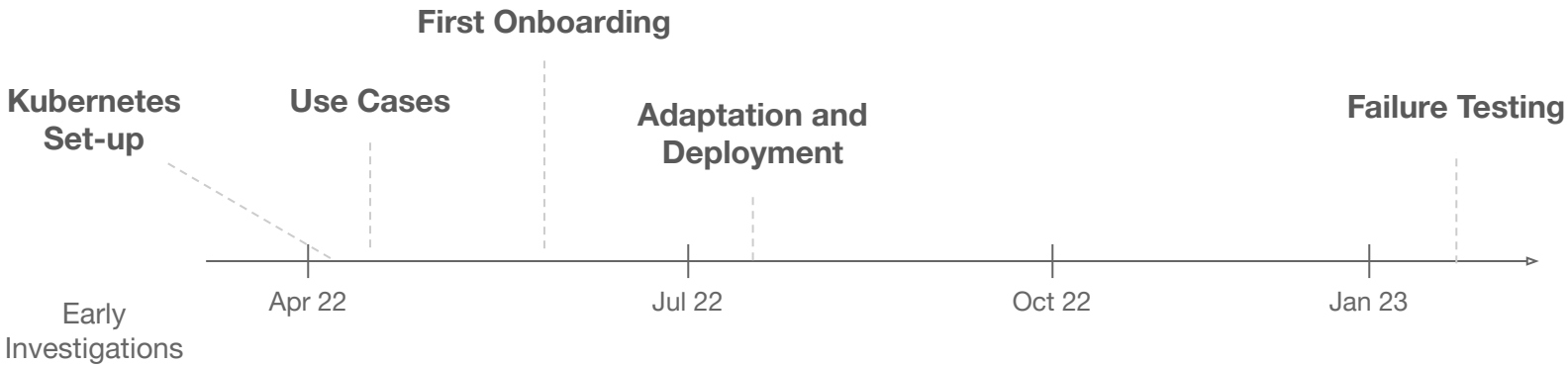
Goals and Drivers

Run all Controls software in an orchestrated, containerized environment before end of LHC long shutdown 3

- Optimize infrastructure resources (cost & energy)
- Improve time to recovery & business continuity
- Streamline DevSecOps/GitOps practices both internally and with industry
- Standardize deployment environments across several data centres

In turn

- Become more agile in hardware & software management
- Reduce costs associated with diverse practices (infrastructure, development, operations, administration)
- Facilitate onboarding & mobility of people working across various sub-systems



Gradual evolution of the infrastructure and functionality.
Second availability zone, artifact replication, support for NFS, LE DNS-01, logs in OpenSearch and more.

Deployment: Principles

Applications and services hosted in the TN / CCR

Setup should survive a GPN disconnection

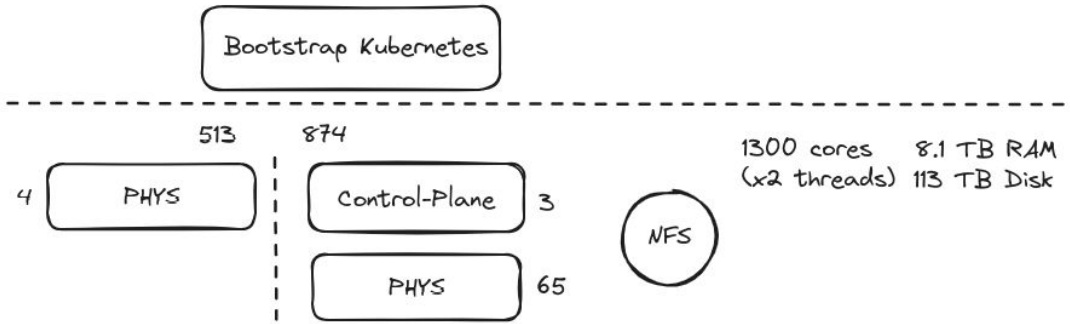
Additional nodes in 513 (TN) for High Availability

No critical dependencies outside the TN

Periodic or GPN-TN synchronization tasks should not prevent normal operation

Deployment: Architecture

Bootstrap Kubernetes cluster to host underlay and registry services



Underlay with a dedicated OpenStack region (tn1), Registry TN

All fully containerised, no dependency on GPN services, DBOD in TN

Region nodes in two availability zones - 513 and CCR

Persistent storage with a NFS backend managed by BE-CSS

Deployment: ArgoCD, GitOps, Alerts

Single place for all applications (registry, openstack, …) and regions

Auto reconciliation, MR to hardware in seconds, no manual edits

Automated alerts to mattermost and telegram

Notable Differences: OpenStack Services

Hardware installation and monitoring by BE-CSS

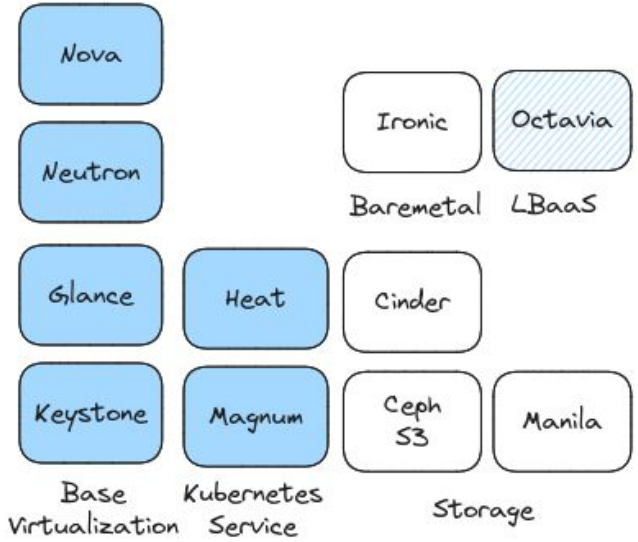
Services required for Virtual Machines and Kubernetes clusters

Prototyped, no commitment for support

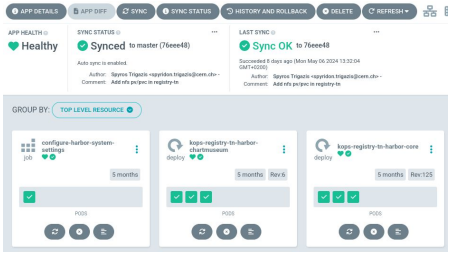
- Octavia / LBaaS, relevant for *serviceType: LB*

Missing vs GPN

- Ironic, Baremetal as a Service
- Cinder, Manila, Swift/*S3*



Notable Differences: OpenStack Support



GPN Service: puppet managed virtual machines

CCR / TN: no puppet service

Solution: fully containerized OpenStack control plane, with Helm / ArgoCD

GPN Service: managed by Server Provisioning (Cloud) FE

CCR / TN: managed by Kubernetes Support FE

GPN Service: project approval by HW Resources FE

CCR / TN: TBD

Notable Differences: OpenStack

OpenStack deployment

- GPN: puppet-managed virtual machines
- TN: ArgoCD applications

OpenStack Support

- GPN: Cloud Infrastructure FE
- TN: Kubernetes FE

Project Approval/Management

- GPN: HW resources FE
- TN: Cloud grants access to TN1 for existing projects

Notable Differences: Persistent Storage

GPN Service: multiple options (Manila / CephFS, Cinder / RBD)

- Dependency on the CEPH backend

CCR / TN: BE-CSS offering of NFS shared storage

Solution: support added for csi-nfs in the Kubernetes Service

- No difference for workload definition, PVs are abstracted
- Also available in the generic GPN service if any workload requires

Notable Differences: Monitoring & Logging

GPN Service: integration with IT MONIT for logging and metrics

Note! Being updated for the new monitoring architecture

CCR / TN: monitoring with COSMOS, managed by BE-CSS

Solution(s)

Logging: direct integration of fluentd with OpenSearch

Also an option for the GPN service from now on

Metrics: work in COSMOS to integrate with Prometheus / OpenTelemetry

Notable Differences: Topology

GPN Service: ongoing work with OpenDCIM integration for improved scheduling

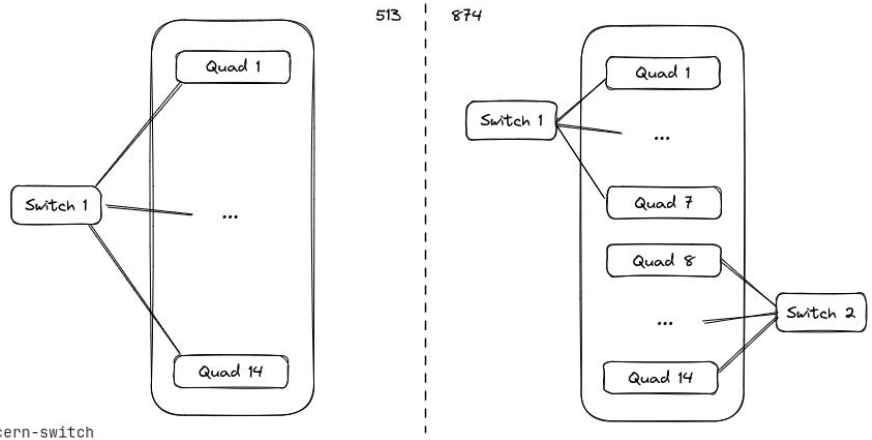
CCR / TN: no easy source for infrastructure topology information

Solution: CCR information integrated with OpenDCIM

New opendcim-topology component

Scheduling aware of AZs but also:

router/service, switch, rack, quad




```
kubectl get no -L topology.kubernetes.io/cern-pdu,topology.kubernetes.io/cern-service,topology.kubernetes.io/cern-switch
NAME                                STATUS   ROLES    AGE   VERSION   CERN-PDU      CERN-SERVICE   CERN-SWITCH
mytestcluster-2ihf6zqlmndy-master-0   Ready    master   14d   v1.29.2   PDU-CD18-01   513-C-IP44     N513-C-IP44-LBR7T-17
mytestcluster-2ihf6zqlmndy-node-0     Ready    <none>   14d   v1.29.2   PDU-BC09-01   513-A-IP560    N513-A-IP560-LBR7T-10
```

Notable Differences: Registry

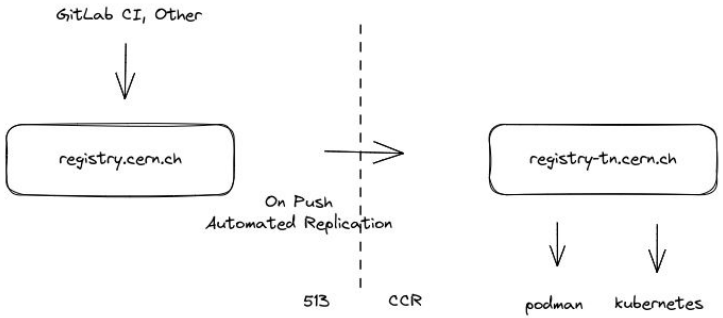
Dedicated replica instance in the CCR / TN, critical path component

GPN Service: S3 / CEPH backend for artifacts

CCR / TN: S3 service currently available, but TN exposed

Solution: S3 backed by Minio on top of NFS

- Automated replication GPN-TN for selected ATS projects, no direct push
- Image curation workflow
- Integrated artifact replication for ACC



Notable Differences: Let's Encrypt Certificates

GPN Service: certificate automation via the ACME HTTP-01 Challenge

- Requires a specific landbset allowing incoming traffic from LE endpoints

CCR / TN: no possibility for any sort of incoming connectivity, no HTTP-01

Solution: support added for the DNS-01 challenge (TXT records)

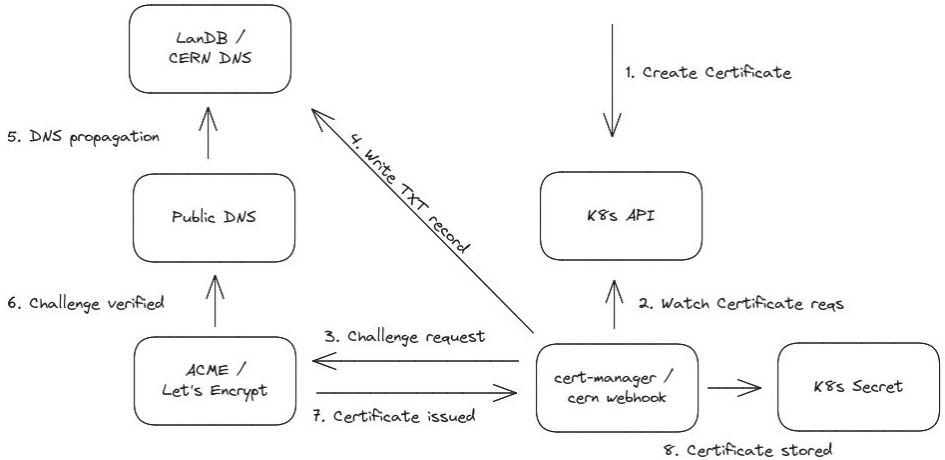
- Also be available for the GPN service
- Requires TN outgoing connectivity for challenge request to the LE servers

Notable Differences: Let's Encrypt Certificates

cert-manager-webhook to handle TXT records in the CERN DNS

Relies on the new REST API to landb, ongoing review with security team

kops-proxy with policy allowing LE challenge request from TN

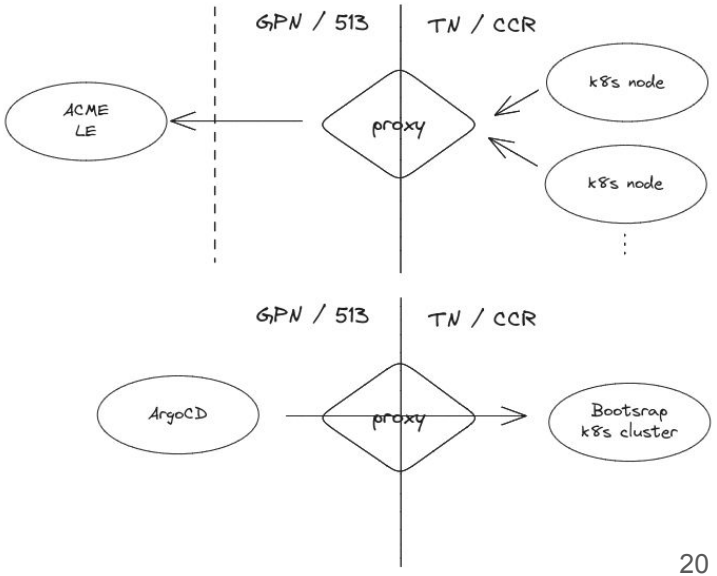


Notable Differences: Proxies

Password protected squid proxies, approved by CERN security team

Two different use cases, monitored by the security team at network level

- RQF2454338: LE Challenge Request
- RQF2454338: ArgoCD Kubernetes Support



Failure Scenarios

April 17th 2024: Registry TN resilience, power cycle nodes to simulate failures

- API degraded but not down (DNS LB), storage is the only point of failure
- Improvements: IP based Load Balancing, HA for storage (alternatives TBD)

TBD May/June: Kubernetes Service

- Either with a real BE-CSS service or a simulated deployment

Summary

Kubernetes deployment fully aligned with the GPN service

- All services managed in a central ArgoCD instance
- Last two service releases already done simultaneously for GPN and CCR/TN

BE-CSS onboarded into service operations (ArgoCD applications, Registry, …)

Almost functionally equivalent deployments

- Missing: *serviceType: LB*, multi-master clusters, *baremetal provisioning*

Open Items

Storage, the current NFS is lacking features (strong auth / authz)

Networking, serviceType: LB and alignment with ATS-IT evolution of network

Octavia?, Multi datacenter HA, Floating IPs, …

OpenStack, identity and general support

Virtualization, is it a requirement?

SSO failure scenarios

Goal: evolve the Kubernetes service so it meets needs from both setups

References

ATS-IT Kubernetes PoC Phase 1 - Final Report

ATS-IT Kubernetes Phase 2 - Milestone Plan

https://kubernetes.docs.cern.ch

Q & A

Backup Slides

Project and General Objectives

Deploy IT managed Kubernetes in the TN (in CCR)

- Validate CCR based setup, discover limitations
- Develop additional integrations where needed

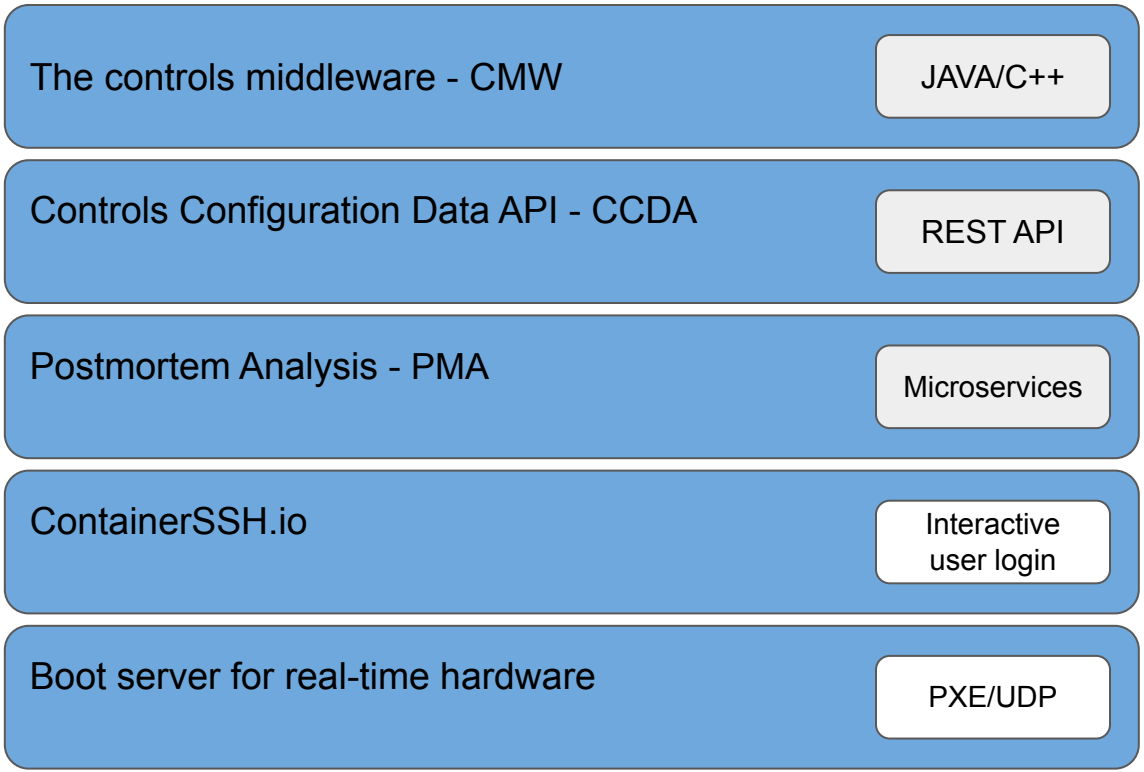
Run ATS controls software on it

- Understand changes required to software, deployment and monitoring tools

Propose next steps to transition towards production

V. Baggiolini, K. Ben Othman, B. Copy, JB. De Martel, F. Ehm, D. Gaponcic, S. Gennaro, R. Gorbonosov, D. Guerra, E. Grancher, E. Genuardi, K. Kessler, N. Laurentino Mendes, T. Oulevey, S. Page, A. Podkowa, R. Rocha, C. Roderick, G. Simonetti, S.Trigazis, M. Treu, K. Turvas, M. Vanden Eynden, R. Vasek, M. Voelkle, R. Voirin

Pilot Services: Overview



Lessons Learned

Frequent communication and sync is key

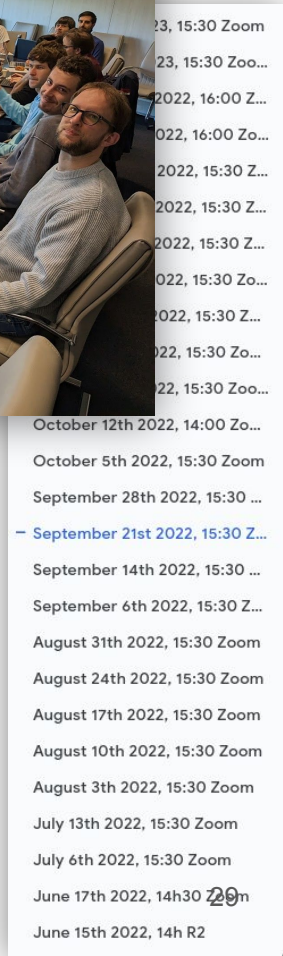
- Weekly meetings very well attended
- Design reviews, office in 774 for IT colleagues

Worlds apart but they do not need to be

- Requirements not distinct enough to justify the differences
- We can do a much better job by working closely together

Steep learning curve is flattening

- New engineers and students arriving with the required knowledge
- Visible reduction in the effort to adapt+deploy applications vs 5 years ago



Phase 1: Failure Scenarios

Registry availability including GPN-TN replication

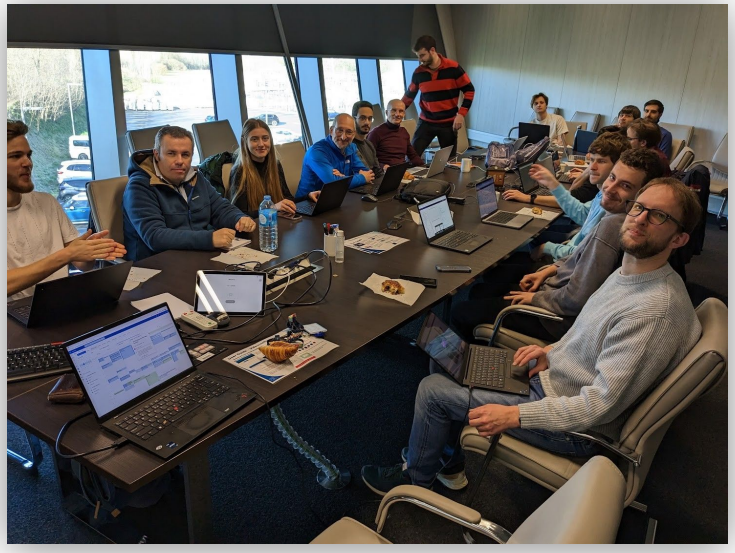
OpenStack and Kubernetes control plane, node failure and restart

Stateless applications, cluster in 1 and 2 AZs

- Single and multi node downtime
- Master downtime

Stateful applications, TBD

Storage topology and availability, TBD



Phase 1: Results

Successful deployment of ATS use cases on Kubernetes

- 5 use cases onboarded in Phase 1, more being worked on
- 18 clusters, 122 nodes across both 513 and the CCR

Extensive Phase 1 Report published covering infrastructure and applications

- Useful for anyone moving from legacy to Kubernetes

Demonstrated multi-datacenter deployments with improved HA

All new developments integrated into the general IT Kubernetes Service

Phase 1: Pilot Service Experiences

Standardization of tools, reduction of in-house solutions

Improved workload isolation with microservices

Easier and more independent upgrades

Rolling upgrades, improvements in the overall speed of application deployment

Steep learning curve

Complexity brought by the additional layers

Integration with existing debugging and troubleshooting tools

Phase 1: Impact

ATS confirmed decision to transition towards Kubernetes and Cloud Native

- Organisation wide commitment for transition to production

Convergence of computing environments between ATS and IT

- Consolidated infrastructure between 513 and 874/CCR, common building blocks
- Much easier to deploy IT services in the CCR and ATS services in 513

Expanded synergies between ATS and IT teams

- New opportunities for further collaboration, teams speaking the same language
- Increased levels of trust from both sides