



OpenShift @ TGW

Full-stack Deployment von Hardware, OpenShift,
TGW Applikationen und kundenspezifischen
Anpassungen in Kunden-Infrastrukturen

What (not) to expect

- This is not a „best practices“
 - try to describe the environment we're working in,
 - its challenges,
 - how we try to mitigate / solve them
- If any questions come up please just raise them
 - If small enough I'll try to answer inline, worst case we need to postpone to Q&A / discussion at the end
- I'm very interested in any recommendations, remarks or ideas
 - We definitely left room for improvement



About me

Klaus Linzner

- Software Architect @ TGW
- Strong Background in .NET – working on base .NET libs for years, some C# talks at conferences
- Couple years ago heading more into build & delivery mechanisms
 - Lead migration to git and azure devops
- Delivery mechanisms and builds were extended to software delivery to site
 - Lead on internal project ADP „Advanced Delivery Platform“ – the project that eventually introduced OpenShift



An aerial photograph of a large industrial facility, likely a warehouse or distribution center. The facility has a long, white, rectangular building with a flat roof. In front of the building is a large paved parking lot. On the left side of the parking lot, numerous semi-trucks are parked in neat rows, facing the building. In the center of the parking lot, there are large stacks of white, rectangular objects, possibly pallets or containers, arranged in a grid-like pattern. On the right side of the parking lot, there are more stacks, but these are red and appear to be made of a different material, possibly bricks or tiles. The facility is surrounded by green trees and grass. A large red shape, resembling a stylized arrow or a drop, is overlaid on the right side of the image, pointing towards the text.

WHAT WE'RE DOING



warehouses

**millions of
crates**

**hundreds of
devices**

hundreds of times



zalando PUMA®



th.mann ESPRIT



STRAUSS

KÄRCHER



VICTORIA'S SECRET

MANGO



what we're doing

a single device can be...

a storage rack



what we're doing

a single device can be...

a shuttle block



what we're doing

a single device can be...

a (part of a) conveyor



what we're doing

a single device can be...

an AMR / AGV



what we're doing

a single device can be...

a palletizer



what we're doing

a single device can be...

a pick station



what we're doing

Depending on customer needs we deliver everything from ground up.

Each site is scaled / designed to customer requirements (number of transports, storage locations) which determines compute resources.

But of course: Standardization is a big factor



what we're doing

The workload we're running is usually rather static.

Peak times usually black friday to christmas.

(can go so far: last chance of deployment for the year is middle of october – next time middle of january)



what we're doing

In general, our software should needs to be able to run entirely air gapped.

Some customers require it (at least in theory).

Maybe more valid: They want to be able to run it without external dependencies.



what we're doing

Many (not all) of those components are time-critical!

With those time constraints, our software usually needs to run in the customers datacenter(s).

This means within the customers infrastructure.



what we're doing

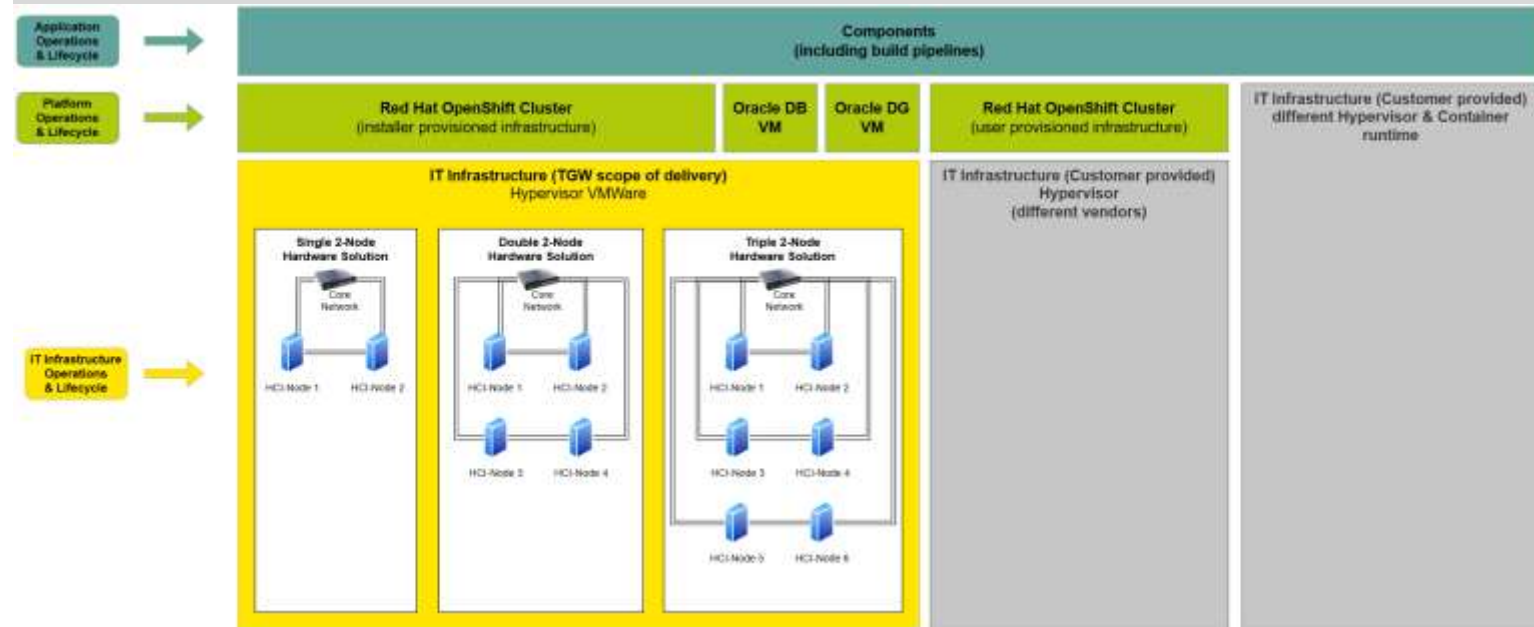
Two approaches to running in customers infrastructure:

Black box (preferred): We ship servers, racks,... As much as possible. And run everything.

Best case: We need a power source and a network cable*

10 years ago this was the default.

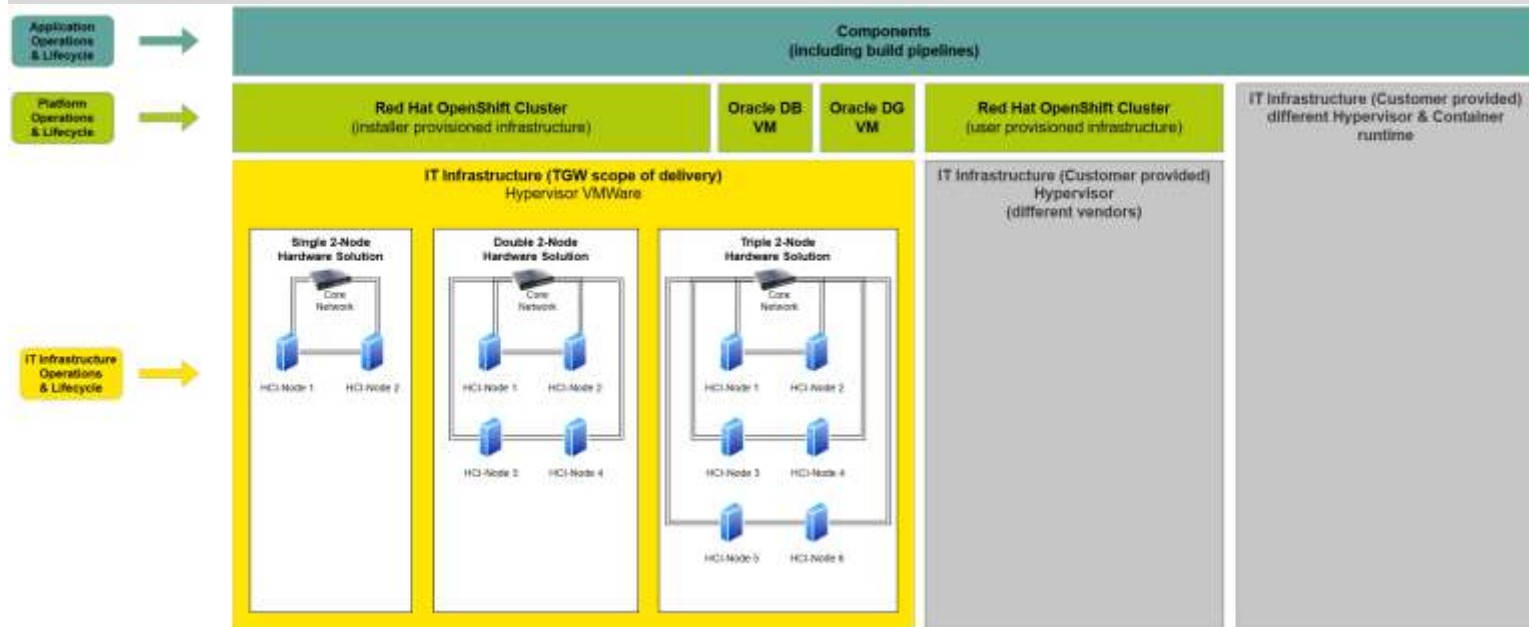
*of course it's not that simple as we're talking about several different VLANs



what we're doing

Grey box: The customer runs hardware, and we run on top of the hypervisor.

As remark: we never ever run directly on the hardware – always a hypervisor in between (hardware patching,...)

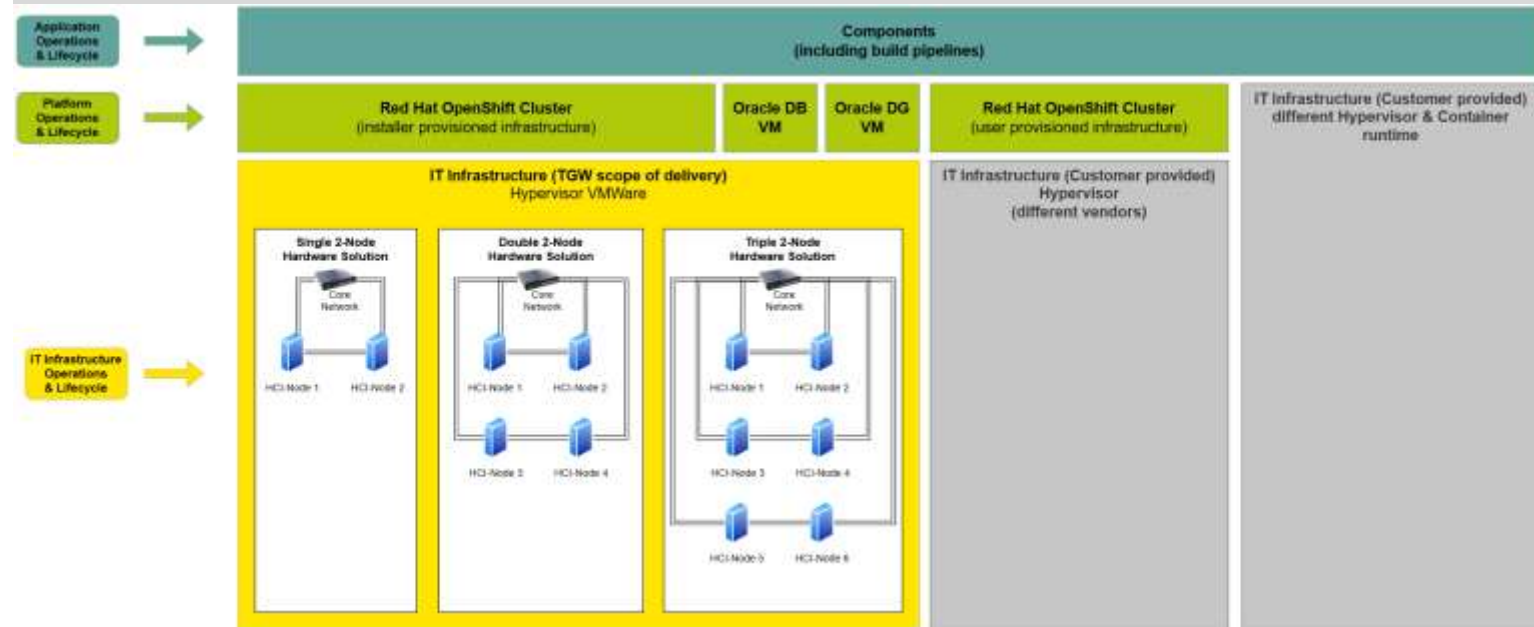


what we're doing

Various shades of grey however:

We ship virtual machine images and customer scans/imports and runs in their environment

Customer provides virtual machines, we install into,



A wide-angle photograph of a modern industrial warehouse or factory floor. The space is filled with complex machinery, including conveyor belts, automated sorting systems, and metal shelving units. A worker in a dark t-shirt with a white 'CI' logo is visible in the middle ground, standing near a conveyor belt. The floor is light-colored, and the ceiling is high with numerous fluorescent lights. A large, semi-transparent red shape is overlaid on the left side of the image, containing the text.

how we came to openshift

Back in 2022

Obviously - all of those devices and their controlling software needs to work together smoothly.

What was NOT obvious for a long time – that all of the software should follow same patterns for deployment – making it easier to use and to update



Back in 2022

The basic software products are developed by different teams.

In different programming languages.

With different frameworks.

Different patterns.

Order of magnitude: 10 teams - 100 developers.
Mostly Austria & Germany



Back in 2022

Product teams ship software components to realization teams.

Some software needs only configuration.

Some software needs customization (customer specific code).

Realization teams integrate the different software parts.

Build it with different tools.

Deploy it with different mechanisms.

Order of magnitude: lots of changing Teams, 200 developers
Worldwide

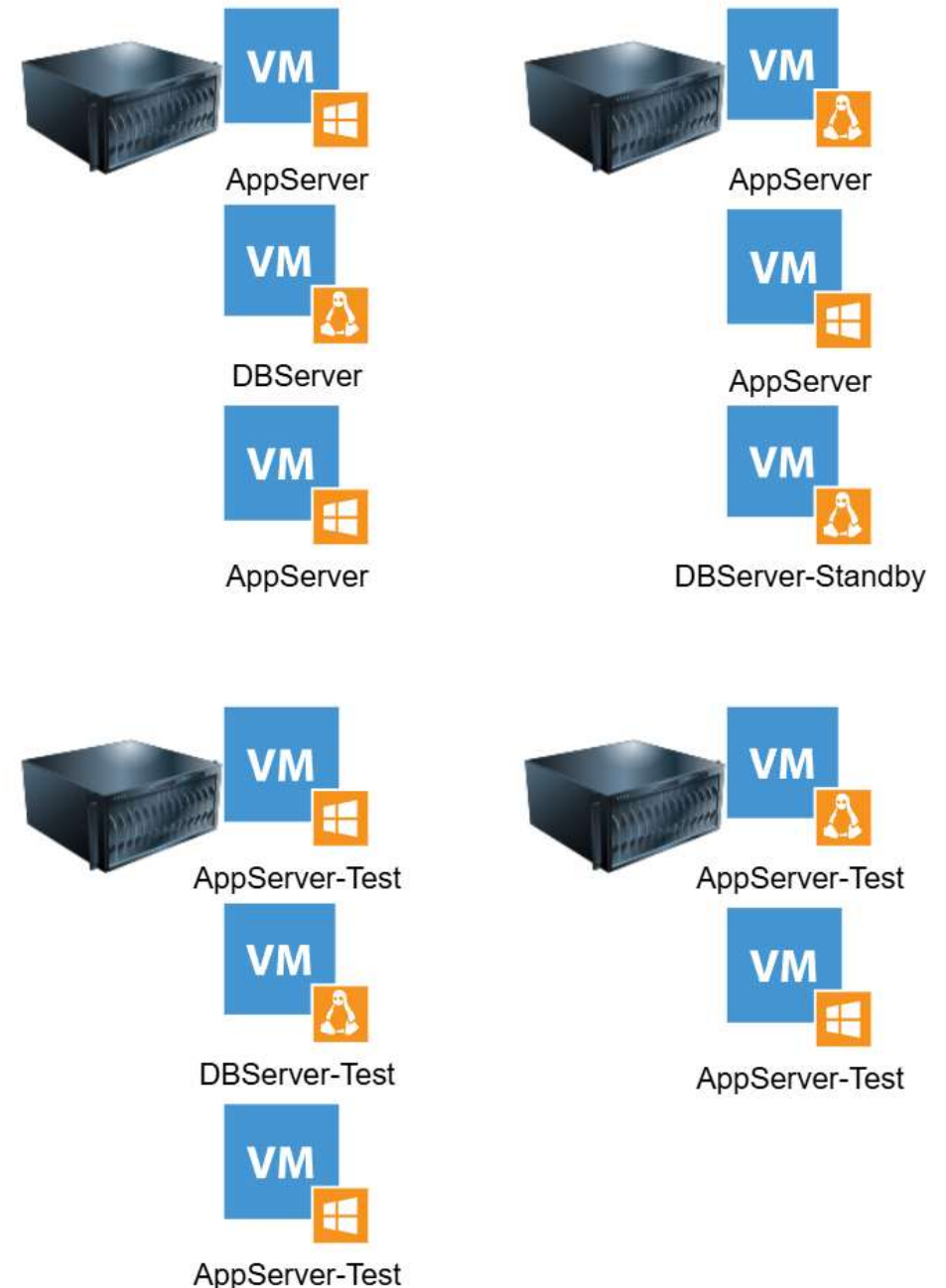


Back in 2022

Every time one of those 300 developers needed a new (in-house) test environment it meant the following:

- Provision 3-5 virtual machines (windows / linux)
- Install required components
- Install required cross-functional services (messaging / authentication)
- Configure all of those
- In some cases: It took a week per test environment (OR: beg your teammates for their environment)

Classic VM Depyoment

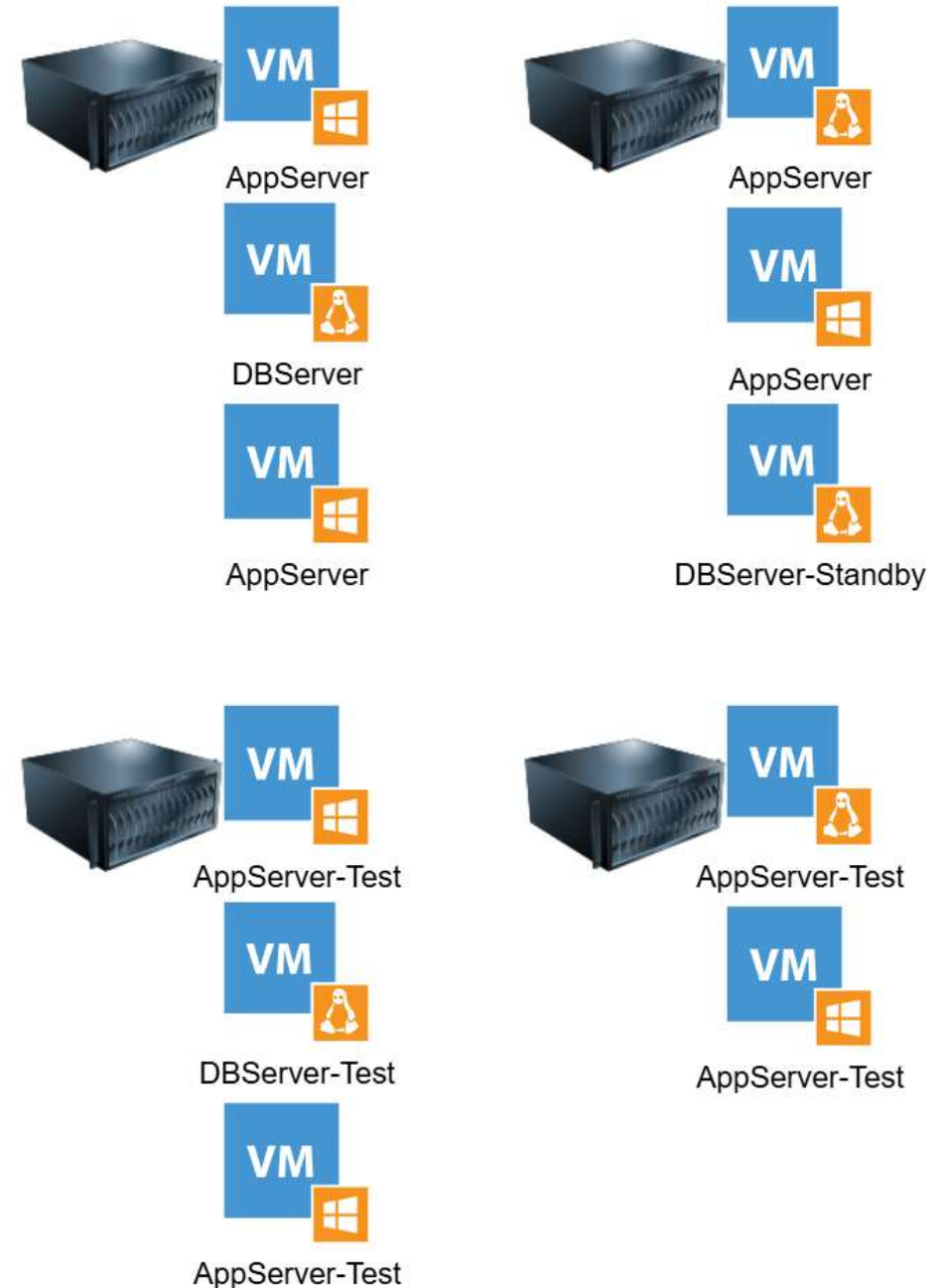


Back in 2022

Every time a developer needed to update...

- Run builds
- RDP/SSH to machine(s)
- Stop service(s)
- Run Installation (different tools)
- Apply (static) configuration
- Start services

Classic VM Depoyoment



Back in 2022

Roughly the same time: We already started consolidating to git / azure devops.

Some components already were there

Some had it on their roadmaps.

What was unclear:

- How do we want to ship software to sites?
- How can we update / patch software / operating system?
- Which features do we need from the system?



Back in 20

Long story short – a year of evaluation and testing.
(In parallel: containerizing applications)

Despite initial reservations kubernetes / OpenShift came up top

- Patching / updating including operating system
- Training material
- Internal DNS / service communication without huge service mesh
- Certificate offloading
- OpenSource

		OpenShift	OpenShift Rating	
		11	12	12
11	Available	11	12	12
12	Available	11	12	12
13	Available	11	12	12
14	Available	11	12	12
15	Available	11	12	12
16	Available	11	12	12
17	Available	11	12	12
18	Available	11	12	12
19	Available	11	12	12
20	Available	11	12	12
21	Available	11	12	12
22	Available	11	12	12
23	Available	11	12	12
24	Available	11	12	12
25	Available	11	12	12
26	Available	11	12	12
27	Available	11	12	12
28	Available	11	12	12
29	Available	11	12	12
30	Available	11	12	12
31	Available	11	12	12
32	Available	11	12	12
33	Available	11	12	12
34	Available	11	12	12
35	Available	11	12	12
36	Available	11	12	12
37	Available	11	12	12
38	Available	11	12	12
39	Available	11	12	12
40	Available	11	12	12
41	Available	11	12	12
42	Available	11	12	12
43	Available	11	12	12
44	Available	11	12	12
45	Available	11	12	12
46	Available	11	12	12
47	Available	11	12	12
48	Available	11	12	12
49	Available	11	12	12
50	Available	11	12	12
51	Available	11	12	12
52	Available	11	12	12
53	Available	11	12	12
54	Available	11	12	12
55	Available	11	12	12
56	Available	11	12	12
57	Available	11	12	12
58	Available	11	12	12
59	Available	11	12	12
60	Available	11	12	12
61	Available	11	12	12
62	Available	11	12	12
63	Available	11	12	12
64	Available	11	12	12
65	Available	11	12	12
66	Available	11	12	12
67	Available	11	12	12
68	Available	11	12	12
69	Available	11	12	12
70	Available	11	12	12
71	Available	11	12	12
72	Available	11	12	12
73	Available	11	12	12
74	Available	11	12	12
75	Available	11	12	12
76	Available	11	12	12
77	Available	11	12	12
78	Available	11	12	12
79	Available	11	12	12
80	Available	11	12	12
81	Available	11	12	12
82	Available	11	12	12
83	Available	11	12	12
84	Available	11	12	12
85	Available	11	12	12
86	Available	11	12	12
87	Available	11	12	12
88	Available	11	12	12
89	Available	11	12	12
90	Available	11	12	12
91	Available	11	12	12
92	Available	11	12	12
93	Available	11	12	12
94	Available	11	12	12
95	Available	11	12	12
96	Available	11	12	12
97	Available	11	12	12
98	Available	11	12	12
99	Available	11	12	12
100	Available	11	12	12
101	Available	11	12	12
102	Available	11	12	12
103	Available	11	12	12
104	Available	11	12	12
105	Available	11	12	12
106	Available	11	12	12
107	Available	11	12	12
108	Available	11	12	12
109	Available	11	12	12
110	Available	11	12	12

Back in 2023/2024

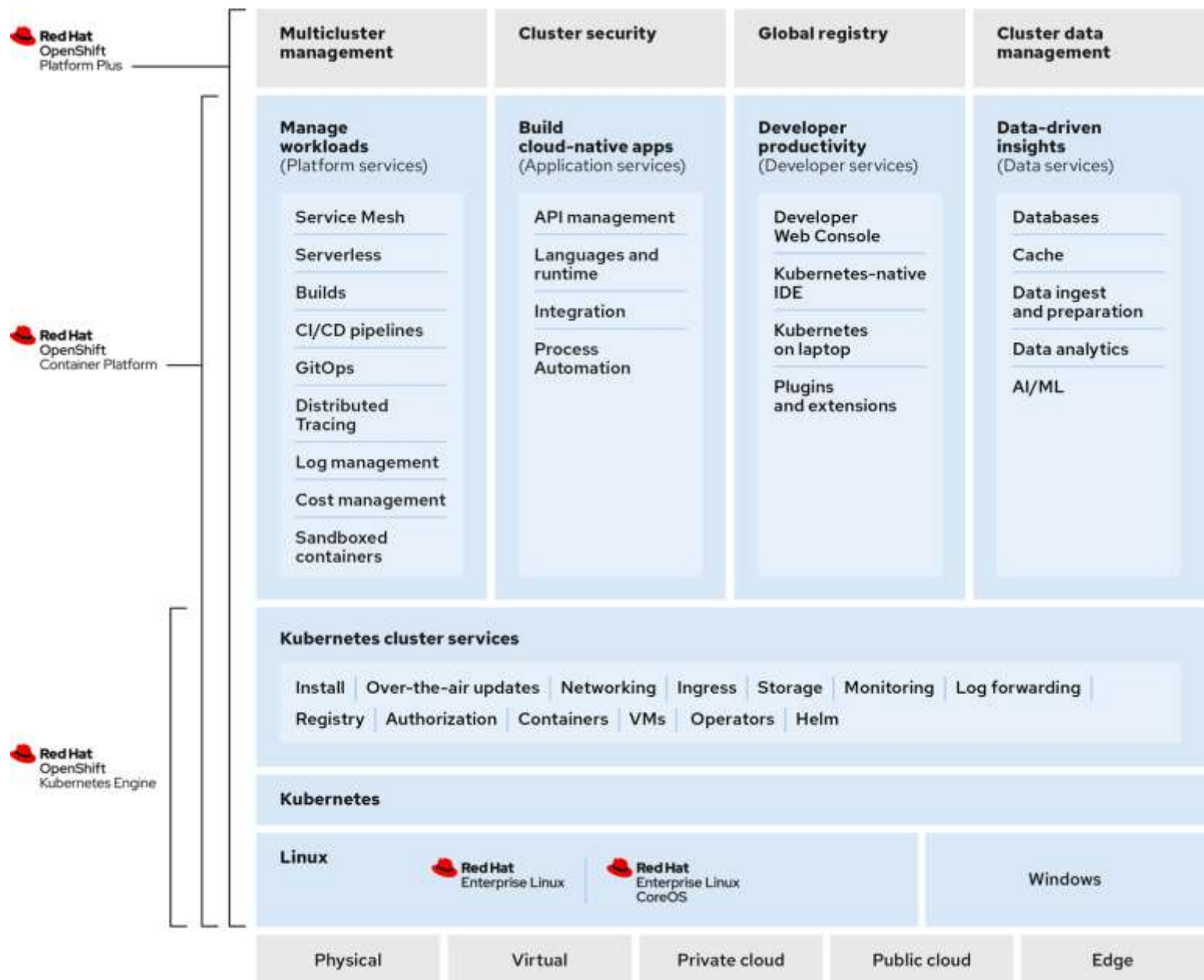
Two notable exceptions / remarks for migration:

- Oracle db remains as-is
- We're using OpenShift Kubernetes Engine

OKE has some downsides as we can't use features provided by OpenShift and that we have to implement ourselves

(we're delivering our own argocd, log aggregation, messaging,...)

But: direct pricing considerations.



The background image shows a large industrial warehouse or factory floor. In the foreground, there are several conveyor belts and automated sorting systems. One conveyor belt is carrying large, flat, brown cardboard boxes. Another conveyor belt further back is carrying black plastic crates. The warehouse has high ceilings with concrete pillars and various pipes and cables running across them. In the background, there are more storage racks and equipment.

INSTALLATION @ CUSTOMER

Installation

Up front: As of September 2025 we installed the new system with 7 customers.

North America committed and started late 2024 with two.

Northern Europe committed soon as well.

Central Europe was on standby and followed once Northern America „was OK“

















After initial concerns regarding our new Platform, teams wanted to migrate additional 15 – but we don't have capacity for it yet

Our Global Locations

Contact us!

With numerous locations on three continents and Headquarters in Marchtrenk, Austria, TGW Logistics has warehouse automation specialists and co-workers available near you. Get in touch with one of our global regions today.

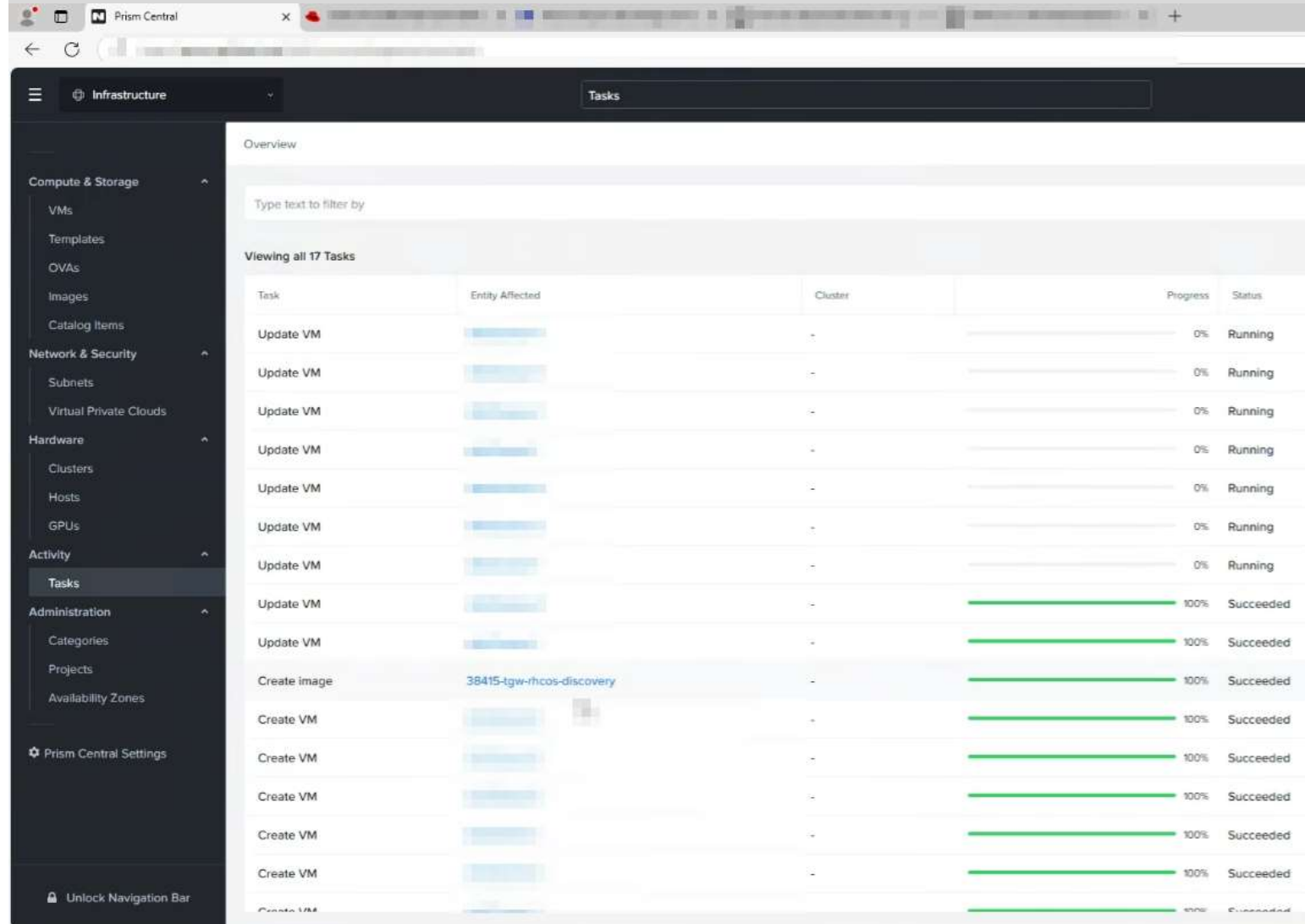


 Austria TGW Logistics GmbH Ludwig Schmitz Straße 3 4614 Marchtrenk +43 50 486	 Austria TGW Logistics GmbH Colbmannstraße 2 Boschstraße 52 4603 Wels +43 50 486	 Austria TGW Systems Integration GmbH Ludwig Schmitz Straße 3 4614 Marchtrenk +43 50 486	 Benelux TGW Systems BV Stadionstraat 10 4815 NG Breda +31 76 200 1240
 China TGW China Co., Ltd. Room 1103-04, Vanke Centre Riverside, No. 518 Minsheng Road Pudong New Area, Shanghai	 China TGW Logistics Equipment Production (Changzhou) Co., Ltd. Plant No. 3, No. 377 South Wuyi Road Wujin District Changzhou 213100, Jiangsu	 France TGW France SAS 1, av. du Général de Gaulle, Immeuble Le Pascal, Bâtiment A, 6ème étage F-94000 Créteil +33 1 77 01 87 70	 France TGW France SAS 17, av. Didier Daurat, Bâtiment Evickde F-31700 Blagnac +33 5 67 20 37 00
 Germany TGW Robotics GmbH Ing.-Anton-Kathrein-Straße 1- 7, 83101 Rohrdorf 83101 Rohrdorf +49 8031 40890-0	 Germany TGW Software Services GmbH Lindenweg 13 92552 Teusn +49 9671 9216-0	 Germany TGW Systems Integration GmbH Johanna-Kinkel-Straße 3 93049 Regensburg +49 9671 9216-0	 Germany TGW Systems Integration GmbH Sales and Service Offices Robert-Bosch-Straße 11a 63225 Langen +49 6103 934 7610
 Italy TGW Italy S.r.l. Via dei marmorari 68, Spilamberto 41057 Modena +39 059 467132	 Lithuania TGW Lithuania, TGW Limited branch Jonavos g. 60 C, Kaunas LT-44182 +370 67111077	 North America TGW Systems Inc. 3001 Orchard Vista Drive, Suite 300 Grand Rapids, MI 49546 +1 616 888 2595	 Poland TGW Systems Integration Sales Office Poland +48 605 152051

Installation

Of those 7 customers

- All of them provided their own hardware
- 5 running VMWare
- 2 running nutanix
- 1 customer with vmware installation already requested reinstall on nutanix as they are migrating within a year
- 2 already running openshift, proposed to run in their cluster (we declined, not there yet)



Prism Central

Infrastructure

Tasks

Overview

Type text to filter by

Viewing all 17 Tasks

Task	Entity Affected	Cluster	Progress	Status
Update VM		-	0%	Running
Update VM		-	0%	Running
Update VM		-	0%	Running
Update VM		-	0%	Running
Update VM		-	0%	Running
Update VM		-	0%	Running
Update VM		-	0%	Running
Update VM		-	0%	Running
Update VM		-	100%	Succeeded
Update VM		-	100%	Succeeded
Create image	38415-tgw-rhcos-discovery	-	100%	Succeeded
Create VM		-	100%	Succeeded
Create VM		-	100%	Succeeded
Create VM		-	100%	Succeeded
Create VM		-	100%	Succeeded
Create VM		-	100%	Succeeded
Create VM		-	100%	Succeeded

Unlock Navigation Bar

Installation

We've created lots of documentation, cheat sheets,... for realization departments to clear with customer.

Some of the clarifications ~~should~~ need to be made BEFORE contracts are signed.

Some of the clarifications are made close to installation (sometimes a few weeks before installation)

The screenshot shows a Confluence page titled "Clarify with customer" under the "Installation" section. The left sidebar contains a navigation menu with the following items: Documentation, Development & Operations, Installation, Start Here: PreSales Clarification, Clarify with customer (expanded), Clarify DNS, Clarify Installation Method, Clarify Network Configuration, Clarify OpenShift Subscription, Clarify outbound connectivity, Clarify Sizing, Request (Security) Infrastructure, Optional: Request OpenShift Subscription, Prepare Installation, Verify Prerequisites, ADP Installation, Troubleshooting, Component Environment Configuration, Realization, Sales, Core Concepts, Tools & Integrations, Training & Education, Knowledgebase, In Progress, Meeting Notes, and Projects. The main content area of the page has a breadcrumb "Pages / ... / Installation" and the title "Clarify with customer". Below the title, it says "Created by Klaus Linzner, last updated on 18.03.25 • 1 minute read". A yellow warning box contains the text: "Before any of the actual preparation starts, the rough outline of the installation should be cleared with customer. It's NOT required at this point to know all details - but to align on key points and let customers know our questions / requirements". Below this, there are two bullet points: "not relevant if IP 10.11.12.13 or 172.60.13.14 is used - but if outbound internet access is possible" and "not relevant the exact data of vsphere is known - but which installation method has to be done". A list of links is provided: Clarify DNS, Clarify Installation Method, Clarify Network Configuration, Clarify OpenShift Subscription, Clarify outbound connectivity, and Clarify Sizing. At the bottom of the page, it says "Powered by Atlassian Confluence 9.2.5 (Cluster node: 542804)" and the Atlassian logo.

Pages / ... / Installation

Clarify with customer

Created by Klaus Linzner, last updated on 18.03.25 • 1 minute read

⚠ Before any of the actual preparation starts, the rough outline of the installation should be cleared with customer. It's NOT required at this point to know all details - but to align on key points and let customers know our questions / requirements

- not relevant if IP 10.11.12.13 or 172.60.13.14 is used - but if outbound internet access is possible
- not relevant the exact data of vsphere is known - but which installation method has to be done

- Clarify DNS
- Clarify Installation Method
- Clarify Network Configuration
- Clarify OpenShift Subscription
- Clarify outbound connectivity
- Clarify Sizing

Like Be the first to like this

Write a comment...

Powered by Atlassian Confluence 9.2.5 (Cluster node: 542804)

ATLASSIAN

Installation

There's (rather) clear documentation in internal confluence.

Each of those customers we gave the detailed requirements up front:

- Required network connectivity
- Required hypervisor permissions

Each of those customers acknowledged and said it was done.

The screenshot shows a Confluence page titled "OpenShift Installation". The left sidebar contains a navigation menu with the following items: Documentation, Development & Operations, Installation, Clarify with customer, Request (Security) Infrastructure, Optional: Request OpenShift Subscription, Prepare Installation, Verify Prerequisites, ADP Installation, OpenShift Installation, Alternative OpenShift Installation Methods, Post-Installation Configuration, GitOps Installation, Activate OpenShift Subscription, Troubleshooting, Component Environment Configuration, Realization, Sales, Core Concepts, and Tools & Integrations. The main content area is titled "OpenShift Installation" and includes a sub-header "Before you begin". The text states: "Installing the OpenShift cluster is both the hardest and easiest thing to do. From the actual work that needs to be done it's by far the easiest and simplest. But misconfigurations regarding network or DNS that don't work as outlined in Clarified with customer will cause the install. Therefore it's highly recommended to check Verify Prerequisites". Below this, there is a section "Before you begin - what should be ready?" with a list of requirements: "have verified prerequisites" and "have the customers vsphere #PI Access Data at hand". Another section "What you will achieve" states: "OpenShift Cluster will be installed in the environment. This is the biggest leap on the road to ADP, once this is finished and v". Below this, there is a list of tasks: "Creating ignition YAML file with OpenShift cluster configs", "Customize ignition.yaml", "Start installing", "Cluster installation", and "Verify". The final section is "Creating ignition YAML file with OpenShift cluster configs", which includes instructions to download the "openshift-install" script and run a command to generate configs. A terminal screenshot shows the command "curl -s -o /dev/null -w '%{size}n' https://mirror.openshift.com/pub/openshift/v4/clients/ocp/4.17.4/openshift-install" and the output "total 613M". Below the terminal screenshot, there is a list of instructions: "navigate to the folder with openshift-installer (might be different path)", "cd ./binaries/openshift-install/4.17.4/", "make installer executable (if it is not)", and "chmod +x openshift-install".

Pages / ... / ADP Installation

OpenShift Installation

Created by Klaus Linzner, last updated by Alexander Neustadt on 17.04.25 • 4 minute read

Before you begin
Installing the OpenShift cluster is both the hardest and easiest thing to do.
From the actual work that needs to be done it's by far the easiest and simplest.
But misconfigurations regarding network or DNS that don't work as outlined in Clarified with customer will cause the install.
Therefore it's highly recommended to check Verify Prerequisites

Before you begin - what should be ready?
Before you start the installation you should

- have verified prerequisites
- have the customers vsphere #PI Access Data at hand

What you will achieve
OpenShift Cluster will be installed in the environment. This is the biggest leap on the road to ADP, once this is finished and v

- Creating ignition YAML file with OpenShift cluster configs
- Customize ignition.yaml
- Start installing
 - Cluster installation
- Verify

Creating ignition YAML file with OpenShift cluster configs

Download **openshift-install** script from the official site or from our repo and run the following command to generate configs.

```
curl -s -o /dev/null -w '%{size}n' https://mirror.openshift.com/pub/openshift/v4/clients/ocp/4.17.4/openshift-install
total 613M
drwxr-xr-x 1 alex alex 4.0K Jan 13 13:03 configs
-rwxr-xr-x 1 alex alex 613M Jan 13 11:45 openshift-install
```

navigate to the folder with openshift-installer (might be different path)
cd ./binaries/openshift-install/4.17.4/

make installer executable (if it is not)
chmod +x openshift-install

Installation

For the actual OpenShift Installation
we're recommending openshift-install

This works brilliant in-house.

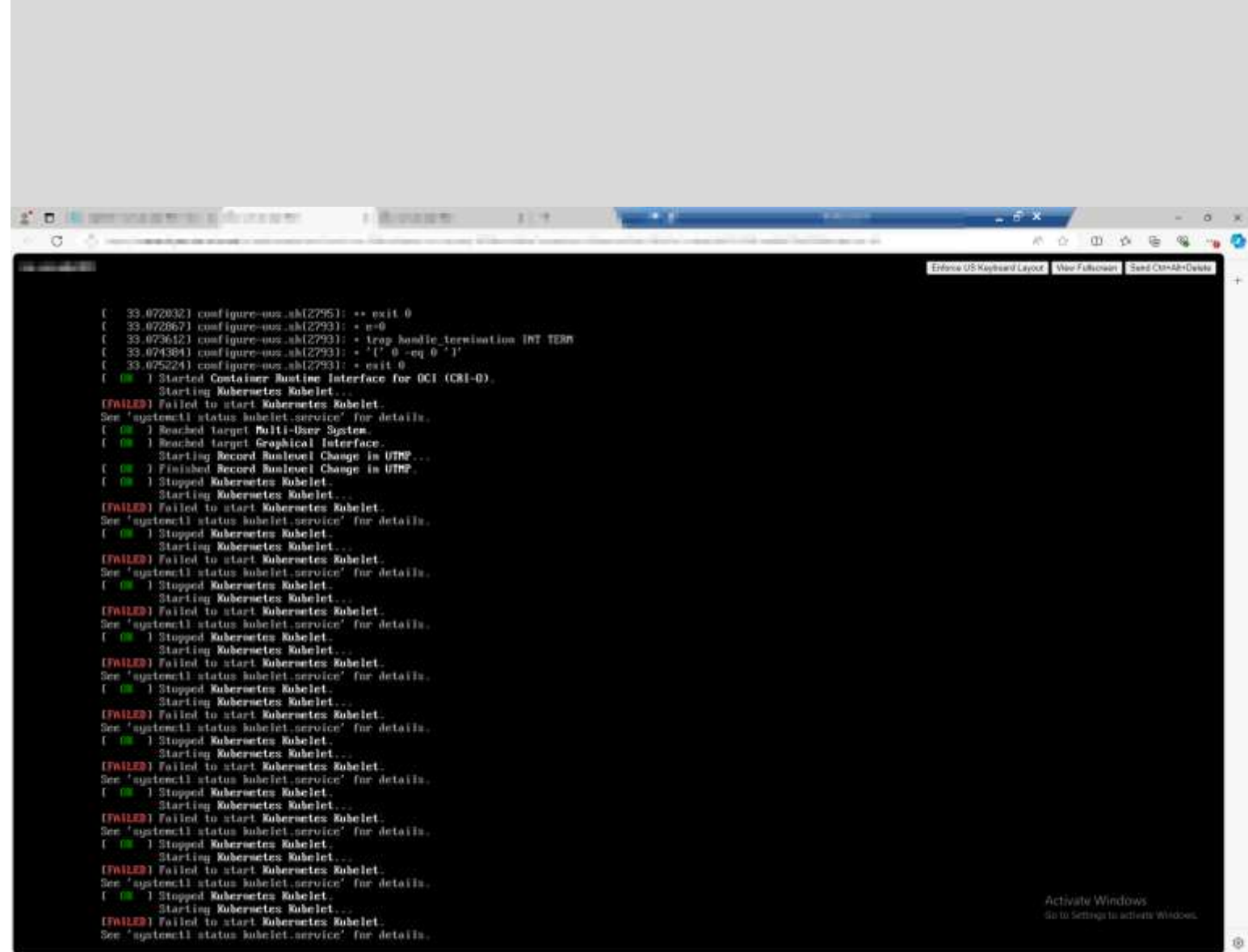
```
[tgw@ ~]$ ./binaries/openshift-installer/4.15$ ./openshift-install create cluster --dir=./config --log-level=info
INFO Consuming Install Config from target directory
INFO Creating infrastructure resources...
INFO Waiting up to 20m0s (until 4:19AM EST) for the Kubernetes API at https://api. ....tgw-group.site:6443...
INFO API v1.28.10+a2c84a3 up
INFO Waiting up to 1h0m0s (until 5:01AM EST) for bootstrapping to complete...
INFO Destroying the bootstrap resources...
INFO Waiting up to 40m0s (until 4:58AM EST) for the cluster at https://api. ....tgw-group.site:6443 to initialize...
INFO Waiting up to 30m0s (until 5:01AM EST) to ensure each cluster operator has finished progressing...
INFO All cluster operators have completed progressing
INFO Checking to see if there is a route at openshift-console/console...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/home/tgw/binaries/openshift-installer/4.15/config/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-console ....tgw-group.site
INFO Login to the console with user: "kubeadmin", ....
INFO Time elapsed: 37m41s
```


Installation

This only worked for 1 customer.

Reasons:

- Network connectivity
- Hypervisor permissions
- Hypervisor connectivity
- Hypervisor internal screwups

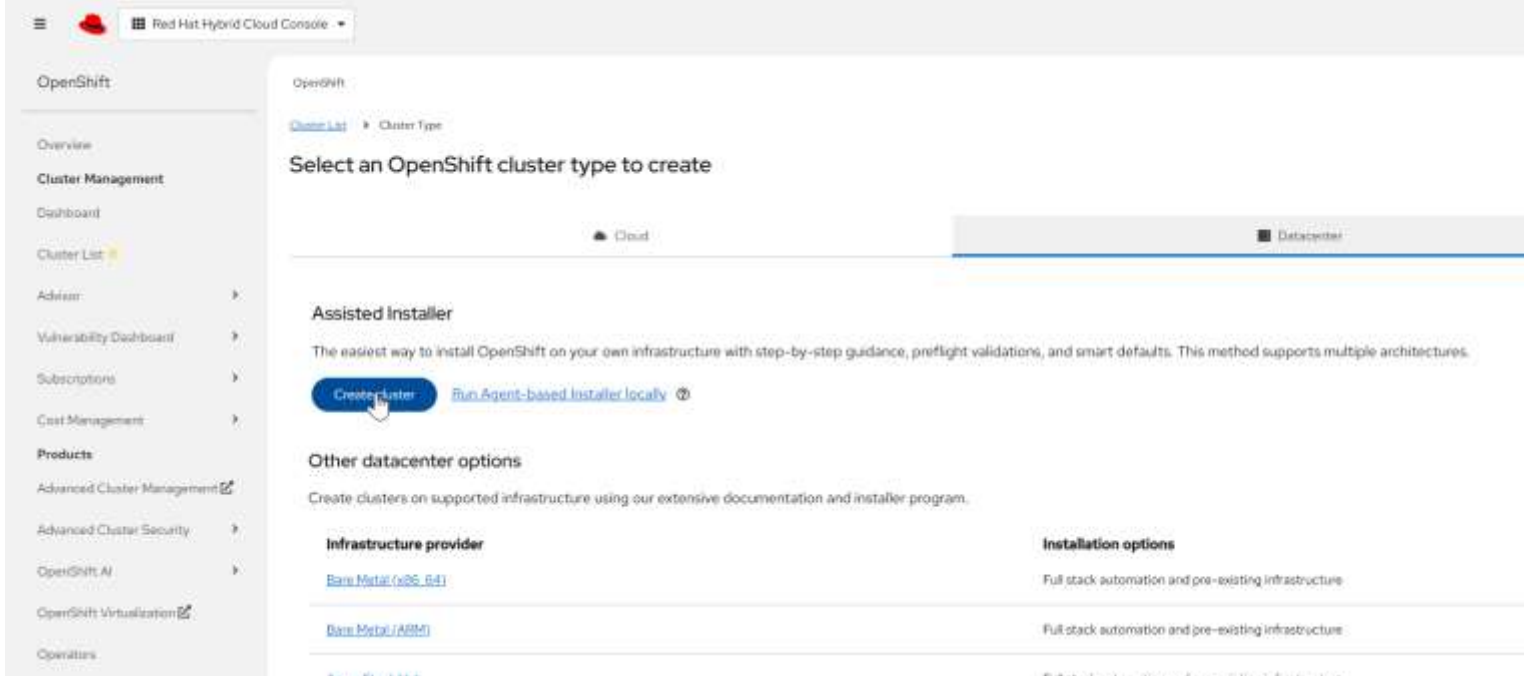


Installation

Fallback to UPI / assisted installer in other cases

Assisted Installer eliminates hypervisor dependencies during installation.

Much more stable / fault tolerant in terms of connectivity (timeouts,...) and more tolerant to resume when errors occur.



Installation

But – even there we had network surprises

(that assisted installer allowed us to work around - MTU default configuration in combination with quay and podman causes installation timeouts)

Installation progress

Started on

4/4/2025, 4:12:38 PM

Failed on 4/4/2025, 4:23:50 PM

[Reset Cluster](#)[Download kubeconfig](#)[View cluster events](#)[Download Installation Logs](#)

Download and save your kubeconfig file in a safe place. This file will be automatically deleted from Assisted Installer's service in 20 days.

Hostname	Role	Status	Discovered on
> ip-10-0-142-10.us-east-1.amazonaws.com	Control plane node	Error 3/7	4/3/2025, 7:19:34 PM
> ip-10-0-142-10.us-east-1.amazonaws.com	Control plane node	Error 3/7	4/3/2025, 7:20:08 PM
> ip-10-0-142-10.us-east-1.amazonaws.com	Control plane node (bootstrap)	Error 4/10	4/3/2025, 7:20:49 PM
> ip-10-0-142-10.us-east-1.amazonaws.com	Worker	Error 4/9	4/3/2025, 7:21:28 PM
> ip-10-0-142-10.us-east-1.amazonaws.com	Worker	Error 4/9	4/3/2025, 7:42:31 PM
> ip-10-0-142-10.us-east-1.amazonaws.com	Worker	Error 4/9	4/3/2025, 7:36:05 PM
> ip-10-0-142-10.us-east-1.amazonaws.com	Worker	Error 4/9	4/3/2025, 7:22:48 PM

Installation

While the product recommendation is openshift-install, the project teams are now heading to assisted installer exclusively.

Easier to install, less code required.

In all cases:

post OpenShift Installation there's one script to install argocd and setup / configure declarative gitops.

```
[root@rhel8 ~]:~/ADP_ClusterConfig/_install_argocd# bash install-argocd.sh
Client Version: 4.17.0-202410241236.p0.gdde885f.assembly.stream.el9-dde885f
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: 4.17.4
Kubernetes Version: v1.30.5
oc tool is installed. Proceeding...

argocd: v2.14.2+ad27246
  BuildDate: 2025-02-06T00:06:23Z
  GitCommit: ad2724661b66ede607db9b5bd4c3c26491f5be67
  GitTreeState: clean
  GoVersion: go1.23.3
  Compiler: gc
  Platform: linux/amd64
ArgoCD CLI is installed. Proceeding...

htpasswd is installed. Proceeding...
git version 2.43.5
git is installed. Proceeding...

No existing gitops or argocd project found. Proceeding...

No existing gitops or argocd subscriptions found. Proceeding...

You are already logged into OpenShift. Proceeding...

Adding password for user okd_admin
Adding password for user okd_user
Warning: resource secrets/htpass-secret is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by oc apply. oc apply should only be used on resources created declaratively by either oc create --save-config or oc apply. The missing annotation will be patched automatically.
secret/htpass-secret configured
Password for okd_admin = [REDACTED]
Password for okd_user = [REDACTED]
Alternatively you can find encoded values in the htpass-secret in the openshift-config namespace

namespace/argocd created

Creating subscription...
subscription.operators.coreos.com/argocd-operator created
```



Installation

In argocd app-of-apps we have

- Cluster config
- Authentication
- Cert-manager installation / config
- Cluster wide daemonsets (logging)
- Prod / pre-prod applications

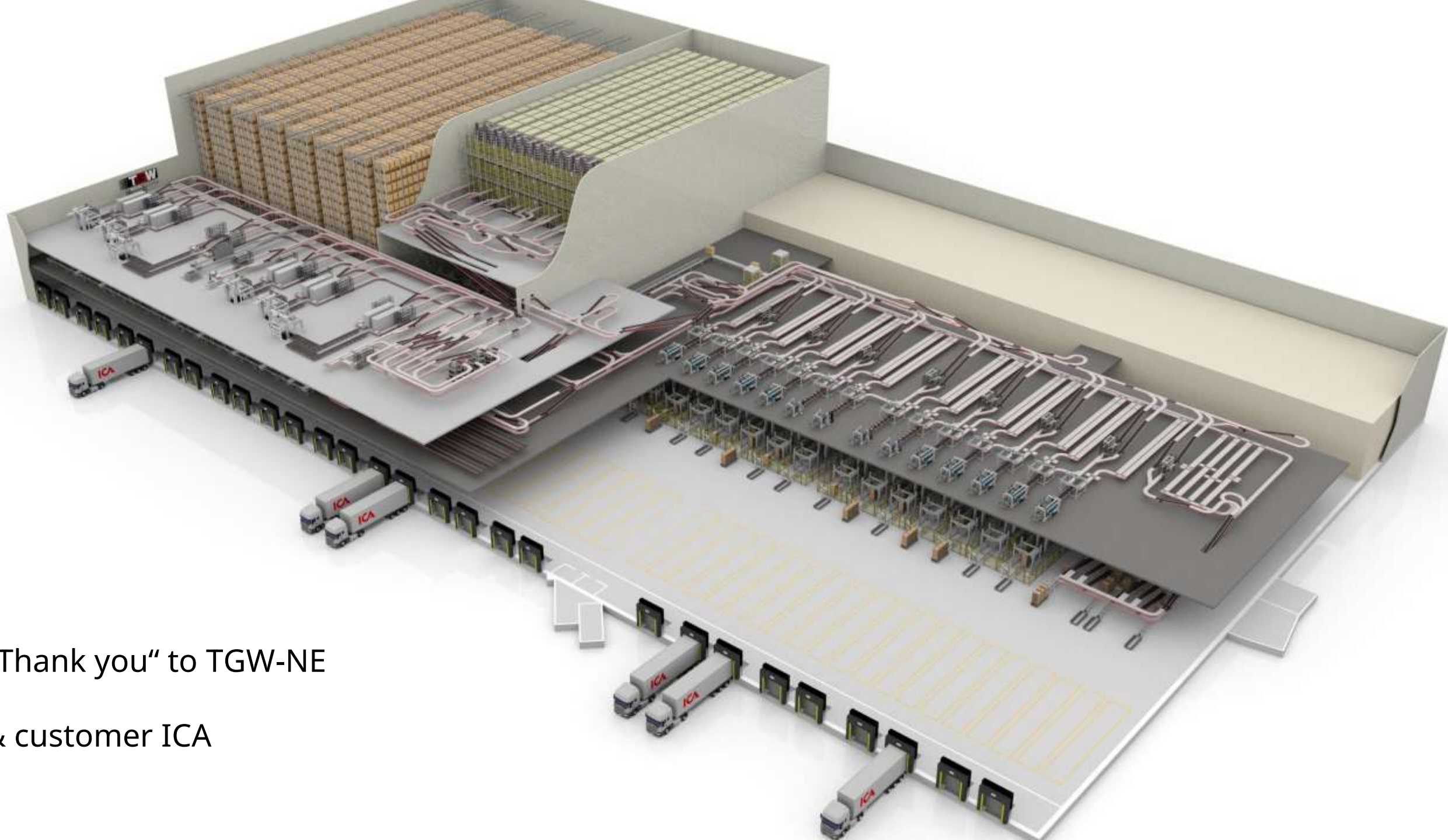
The screenshot displays the ArgoCD web interface. On the left is a dark sidebar with navigation links: Applications, Settings, User Info, and Documentation. Below these are sections for 'Favorites Only', 'SYNC STATUS' (with counts for Unknown, Synced, and OutOfSync), 'HEALTH STATUS' (with counts for Unknown, Progressing, Suspended, Healthy, Degraded, and Missing), and 'LABELS', 'PROJECTS', 'CLUSTERS', 'NAMESPACES', and 'AUTO SYNC'. The main area is titled 'Applications' and features a search bar and buttons for '+ NEW APP', '+ SYNC APPS', and '+ REFRESH APPS'. It shows a grid of application tiles, each representing an ArgoCD application. Each tile includes the application name, project, labels, status, repository, target revision, path, destination, namespace, creation time, and last sync time. At the bottom of each tile are buttons for SYNC, REFRESH, and DELETE. The applications shown are: argocd-cluster-config, cluster-authentication, cluster-cert-manager, cluster-cert-manager-config, cluster-config, kubernetes-secret-generator, werx-infrastructure, werx-warehouse-preprod, and werx-warehouse-prod.

Application	Project	Status	Repository	Target Rev	Path	Destination	Namespace	Created At	Last Sync
argocd-cluster-config	argocd-cluster-config	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	argocd-cluster-config	in-cluster	argocd	12/04/2024 14:23:05 (2 months ago)	12/11/2024 15:04:14 (2 months ago)
cluster-authentication	cluster-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	cluster-authentication	in-cluster		12/04/2024 14:23:12 (2 months ago)	12/05/2024 14:15:12 (2 months ago)
cluster-cert-manager	cluster-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	cluster-cert-manager	in-cluster		12/04/2024 14:23:12 (2 months ago)	12/04/2024 14:28:45 (2 months ago)
cluster-cert-manager-config	cluster-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	cluster-cert-manager-config	in-cluster		12/04/2024 14:23:12 (2 months ago)	12/04/2024 14:29:41 (2 months ago)
cluster-config	cluster-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	cluster-config	in-cluster		12/04/2024 14:23:12 (2 months ago)	12/11/2024 11:30:00 (2 months ago)
kubernetes-secret-generator	cluster-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	kubernetes-secret-generator	in-cluster		12/04/2024 14:23:12 (2 months ago)	
werx-infrastructure	werx-infrastructure-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	werx-infrastructure	in-cluster	werx-infrastructure	12/04/2024 14:23:12 (2 months ago)	01/30/2025 13:16:12 (10 days ago)
werx-warehouse-preprod	werx-warehouse-preprod-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	preprod	in-cluster	werx-warehouse-preprod	12/04/2024 14:23:12 (2 months ago)	
werx-warehouse-prod	werx-warehouse-prod-project	Healthy	https://TGWSoftware@dev.azure.com/TGWSo...	Release/Prod	prod	in-cluster	werx-warehouse-prod	12/04/2024 14:23:12 (2 months ago)	

A wide-angle photograph of a modern industrial warehouse or factory floor. The space is filled with complex machinery, including conveyor belts, automated sorting systems, and metal shelving units. A worker in a dark shirt with a white 'CI' logo is visible in the middle ground, standing near a conveyor belt. The floor is light-colored, and the ceiling is high with numerous fluorescent lights. A large, semi-transparent red shape is overlaid on the left side of the image, containing white text.

how we use openshift & argo

(deep dive into an actual
customer project)



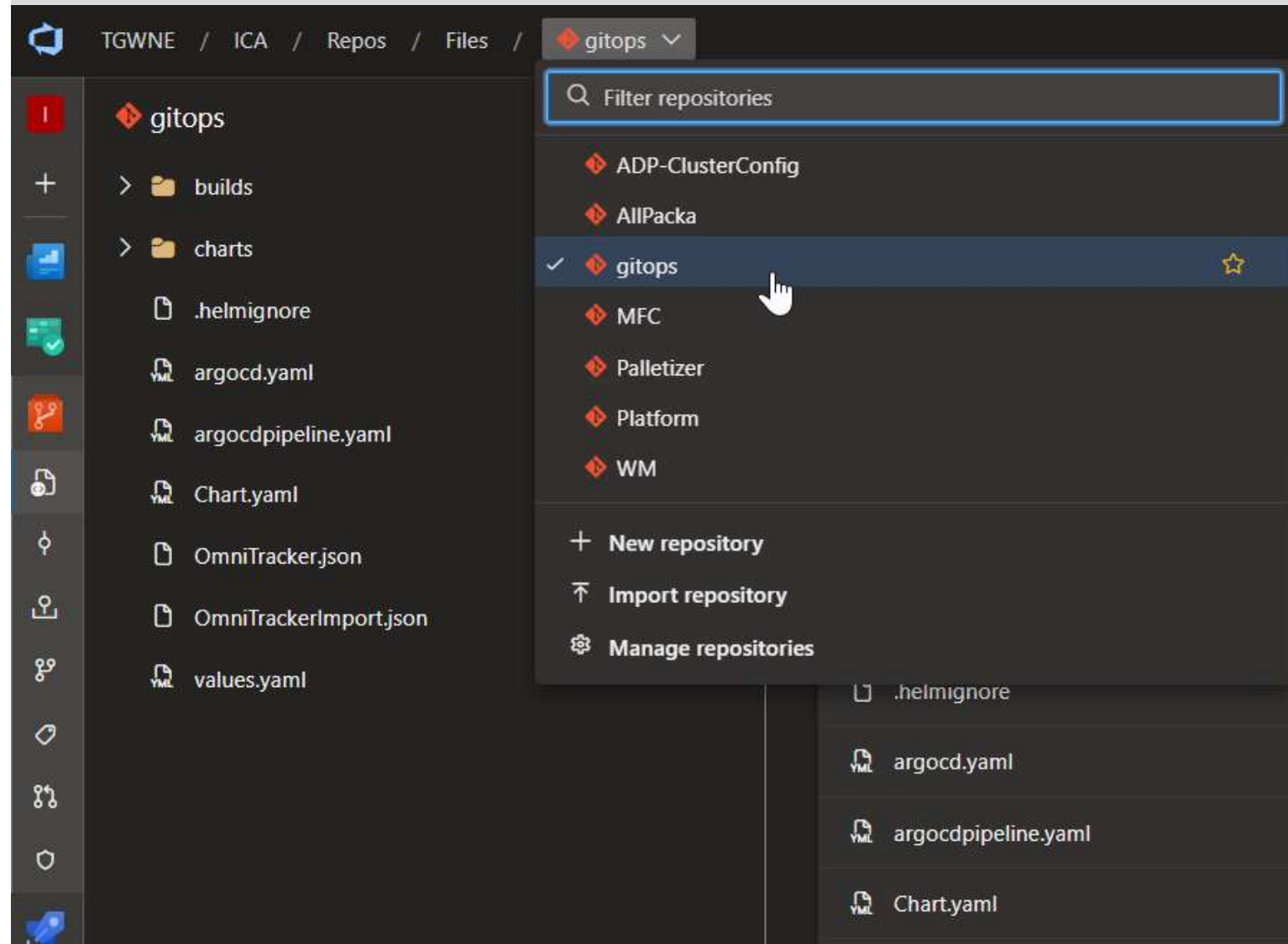
„Thank you“ to TGW-NE
& customer ICA

How we use openshift

Each warehouse / each customer project has its own azure devops project.

In each project there are multiple repos.

- One for each component that is required,
- One „ADP-ClusterConfig“ repo containing,
- One „gitops“ repo containing kubernetes definitions for all applications



How we use openshift

Projects usually start in Azure DevOps:

We have a self service pipeline devs can go to.

- It creates a new devops project...
- clones all desired component / product releases...
- adds azure devops pipelines...
- sets up links / permissions for container registries...
- sets up permissions for repos / build user...
- applies basic configuration (org specific feeds,...)

After ~10minutes project is set up in clean and standardized way.

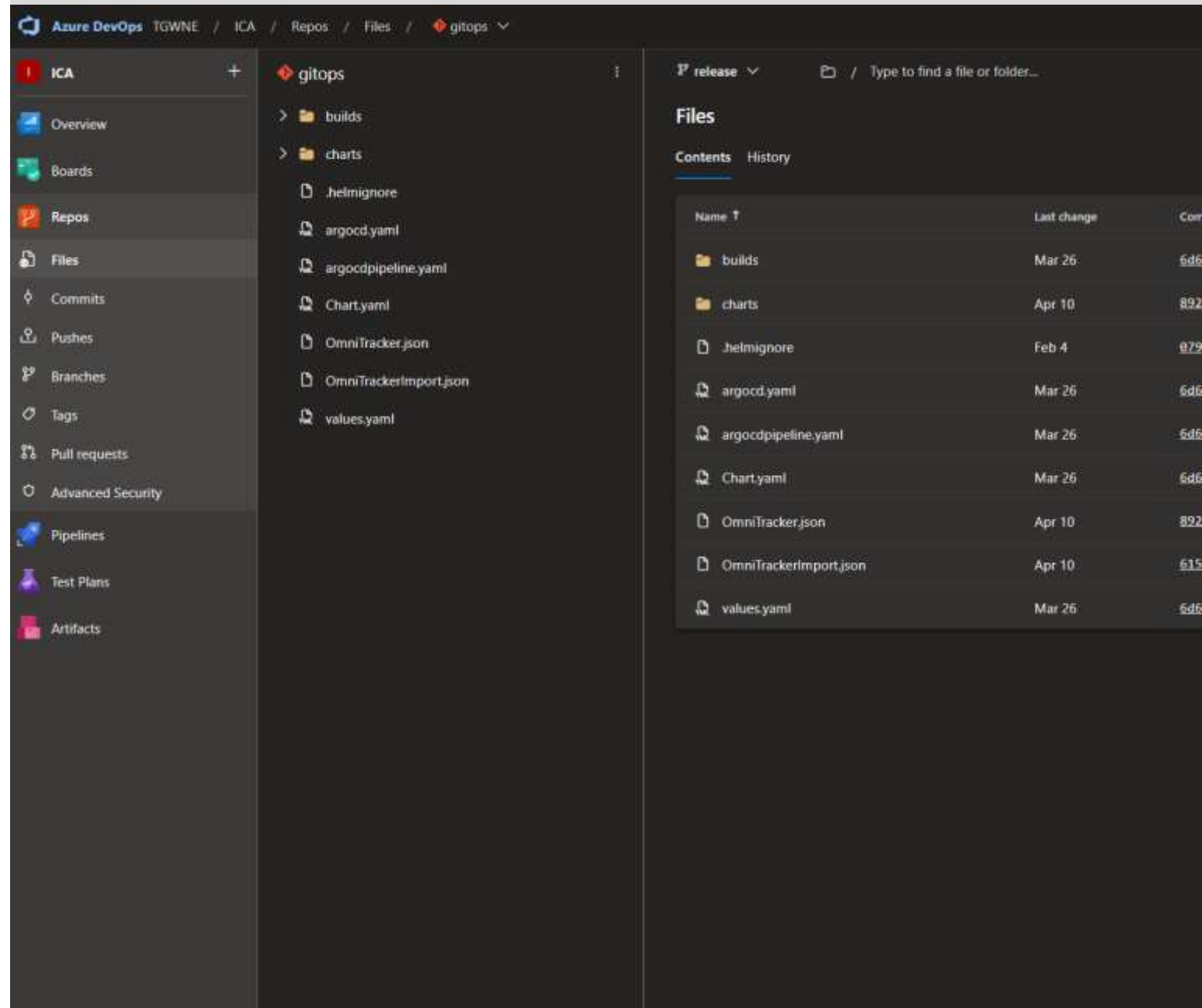


The screenshot shows the 'Run pipeline' dialog in Azure DevOps. At the top, it says 'Run pipeline' and 'Select parameters below and manually run the pipeline'. Below this is a 'Branch/tag' dropdown menu with 'main' selected. Underneath is a section titled 'Parameters'. It contains several dropdown menus: 'Organization' (TGWNE), 'Project' (OpenShiftUserGroup), 'Default Branch in Project' (master), 'ADP-ClusterConfig Release Branch' (Release/ADP-ClusterConfig/v0.3), 'AllPacka Release Branch' (Release/AllPacka/v0.14), 'AVA Release Branch' (<Skip Component>), 'DataDashboard Release Branch' (<Skip Component>), 'ESVT Release Branch' (<Skip Component>), 'MFC Release Branch' (Release/MFC/v8.3), and 'Palletizer Release Branch' (<Skip Component>). The 'Palletizer Release Branch' dropdown is currently open, showing options: '<Skip Component>', 'Release/Palletizer/v1.1', 'Release/Palletizer/v1.0', and 'Release/Palletizer/v0.1'. At the bottom, there is a checkbox for 'Enable system diagnostics' and two buttons: 'Cancel' and 'Run'.

How we use openshift

gitops repo essentially is one large HELM umbrella chart, containing (almost*) all information required to run a site.

*exceptions: production secrets that should not go into source control (oracle connection strings, cert-manager secrets)

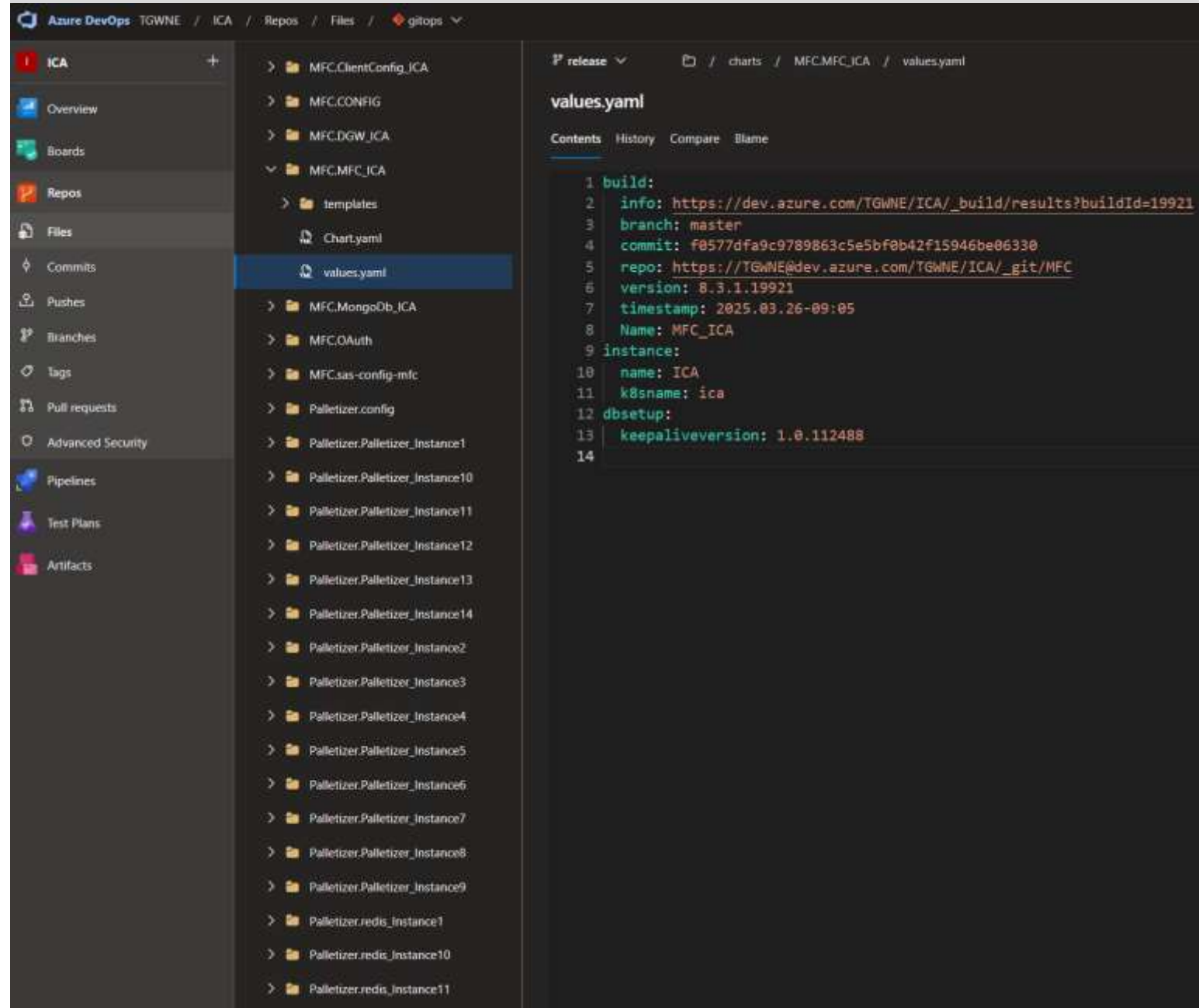


How we use openshift

Ground rules we have that show up in gitops repo

- New deliveries to customer environments can be staged.
- Publishing from stage to production must not rebuild
- None of our cluster is reachable from internet
- We don't „push“ into the customers environment.
Customer is in control - environment is pulling.*

* this helps a lot in customer discussions

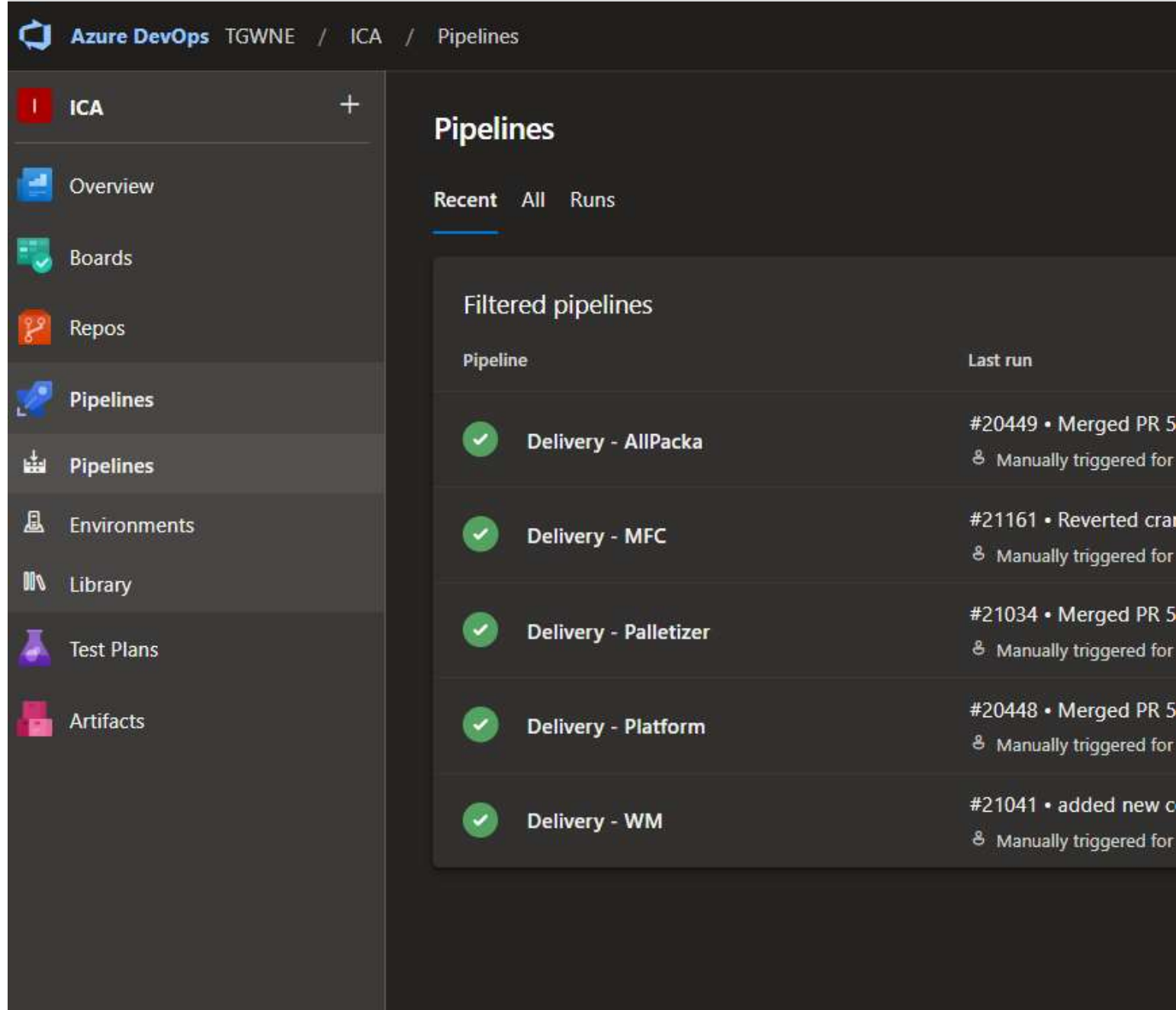


How we use openshift

Gitops repo is not manipulated directly / manually. This happens exclusively through pipelines*.

Each component has a „Delivery“ pipeline

* At least for regular workflows and production use cases. Can be different during development of kubernetes resources



The screenshot shows the Azure DevOps interface for the 'ICA' project. The left sidebar contains a navigation menu with the following items: Overview, Boards, Repos, Pipelines (selected), Pipelines (with a download icon), Environments, Library, Test Plans, and Artifacts. The main area is titled 'Pipelines' and has tabs for 'Recent', 'All', and 'Runs'. Below the tabs, there is a section titled 'Filtered pipelines' which displays a table of pipeline runs.

Pipeline	Last run
✓ Delivery - AllPacka	#20449 • Merged PR 5 ⚙️ Manually triggered for
✓ Delivery - MFC	#21161 • Reverted cran ⚙️ Manually triggered for
✓ Delivery - Palletizer	#21034 • Merged PR 5 ⚙️ Manually triggered for
✓ Delivery - Platform	#20448 • Merged PR 5 ⚙️ Manually triggered for
✓ Delivery - WM	#21041 • added new c ⚙️ Manually triggered for

How we use openshift

Delivery Pipelines...

- Compile source code,
- Embedd static configuration
- Other policies (code analysis, package/license scans,...)
- create SBOM files,
- create container images,
- push their kubernetes definitions into the gitops repo

The screenshot displays the Azure DevOps web interface. The left sidebar shows the navigation menu with 'Repos' selected. The main area shows the 'gitops' repository with a file explorer on the left listing files like 'builds', 'charts', 'helmignore', 'argocd.yaml', 'argocdpipeline.yaml', 'Chart.yaml', 'OmniTracker.json', 'OmniTrackerImport.json', and 'values.yaml'. On the right, the 'History' tab is active, showing a vertical timeline of commits. Each commit entry includes a title, a short hash, the author 'ICA Build Service (TGWNE)', and the commit time.

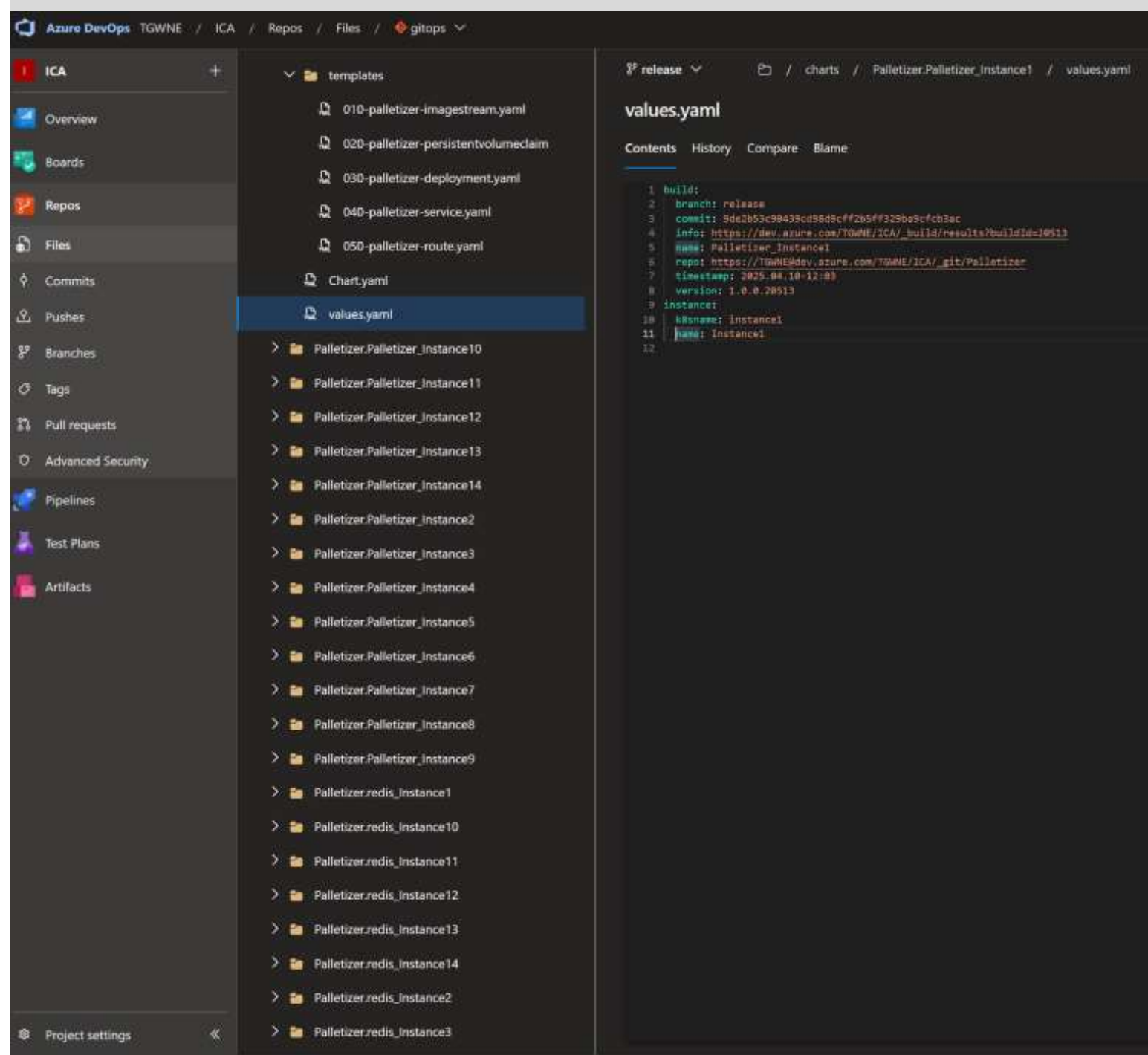
Commit Hash	Commit Message	Author	Time
892e8563	Delivery - Palletizer to 1.0.0.20513	ICA Build Service (TGWNE)	Apr 10 at 2:03 PM
63563432	Delivery - Platform to 1.2.1.20448	ICA Build Service (TGWNE)	Apr 10 at 8:25 AM
8fd8be23	Delivery - AllPacka to 0.9.1.20424	ICA Build Service (TGWNE)	Apr 9 at 3:32 PM
43cb5b05	Delivery - Palletizer to 1.0.0.19958	ICA Build Service (TGWNE)	Mar 26 at 10:56 AM
6daa8046	Delivery - Platform to 1.2.1.19954	ICA Build Service (TGWNE)	Mar 26 at 10:55 AM
4c6c932f	Delivery - AllPacka to 0.9.1.19959	ICA Build Service (TGWNE)	Mar 26 at 10:54 AM
6328d486	Delivery - WM to 10.53.0.19928-SNAPSHOT	ICA Build Service (TGWNE)	Mar 26 at 10:15 AM
954faec	Delivery - MFC to 8.3.1.19921	ICA Build Service (TGWNE)	Mar 26 at 10:06 AM
8143d221f	Delivery - Palletizer to 1.0.0.19911	ICA Build Service (TGWNE)	Mar 26 at 9:58 AM
7b66fdae	Delivery - Platform to 0.4.2.19912	ICA Build Service (TGWNE)	Mar 26 at 9:55 AM
6df9e53d	Delivery - AllPacka to 0.8.1.128672	ICA Build Service (TGWNE)	Mar 26 at 9:55 AM
3e23350e	Delivery - Palletizer to 1.0.0.18877	ICA Build Service (TGWNE)	Feb 4 at 4:15 PM
87936772	Delivery - Platform to 0.4.2.18876	ICA Build Service (TGWNE)	Feb 4 at 4:12 PM

How we use openshift

We use kubernetes horizontal scaling – but not exclusively.

Most of our scaling requirements are bound to the underlying devices.

If we have f.e. 14 palletizer devices running in a customer environment, we have 14 instances running (each one connecting to a specific device and controlling it tightly)

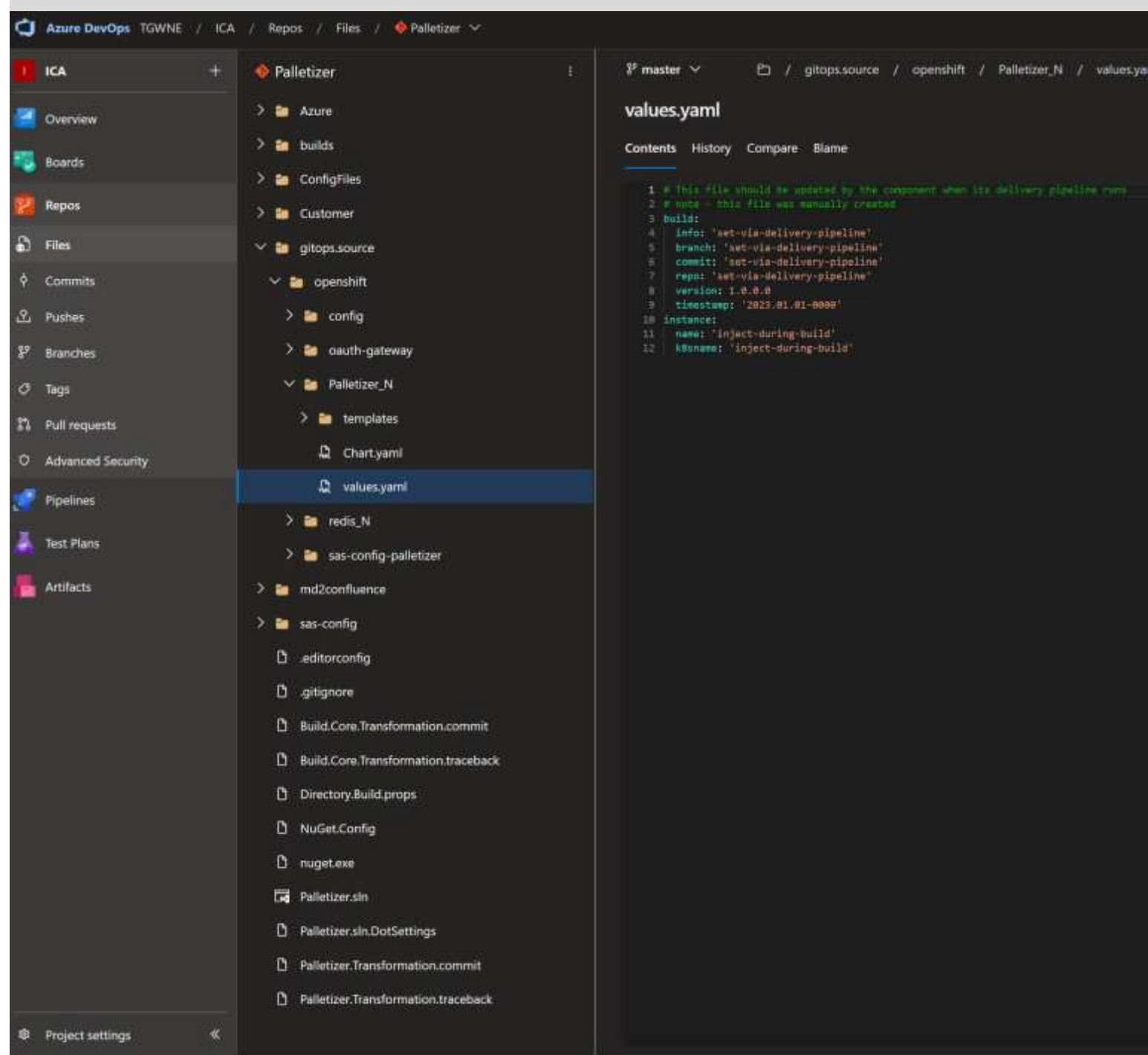


How we use openshift

Internally this is done by the Delivery Pipelines

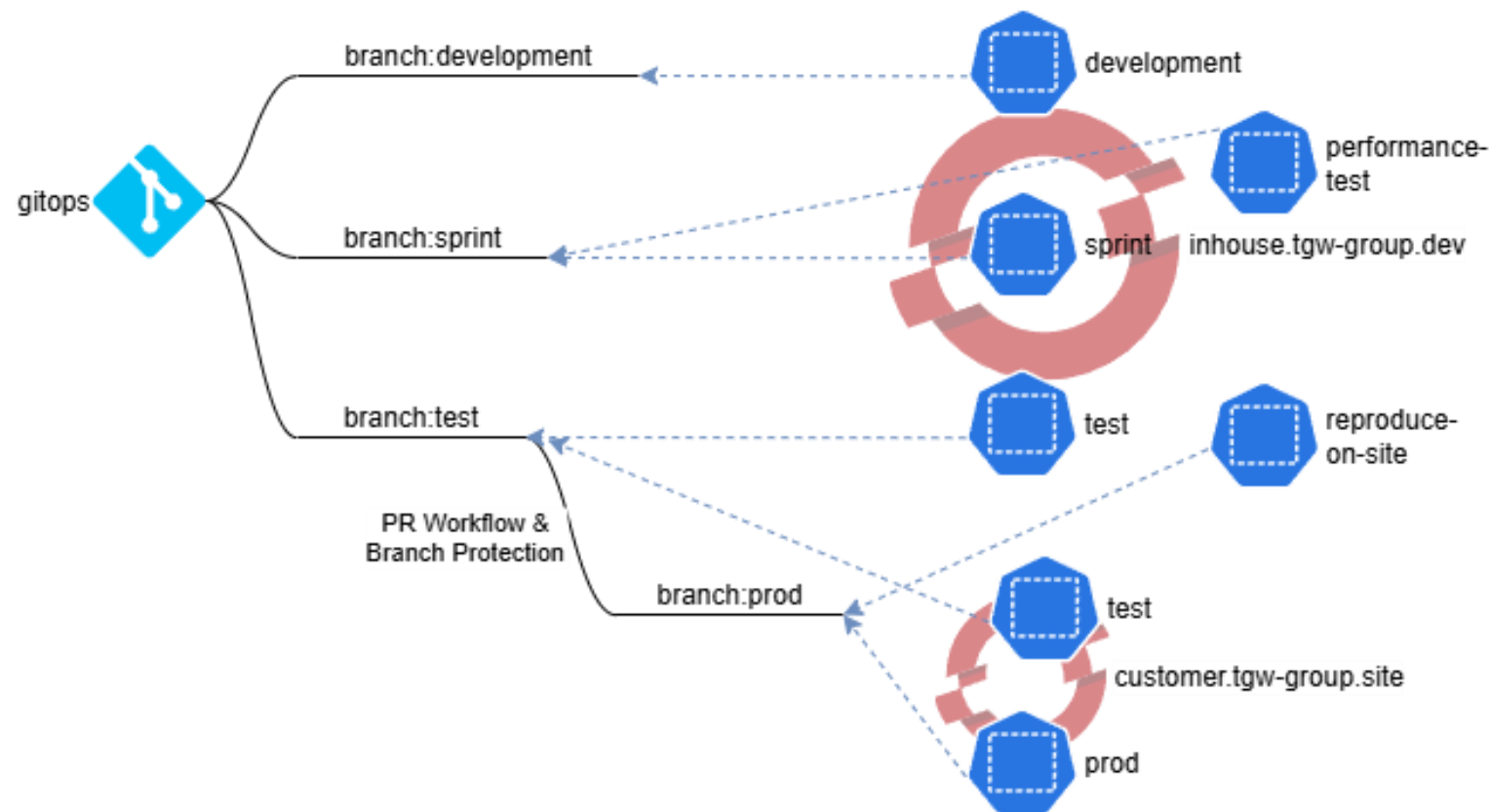
The source components repo has only one single palletizer kubernetes definition. The delivery pipeline checks the config how many instances are configured and expands accordingly.

This required additional effort on our side during pipeline creation, however we can control each device individually (and even update them individually)



How we use openshift

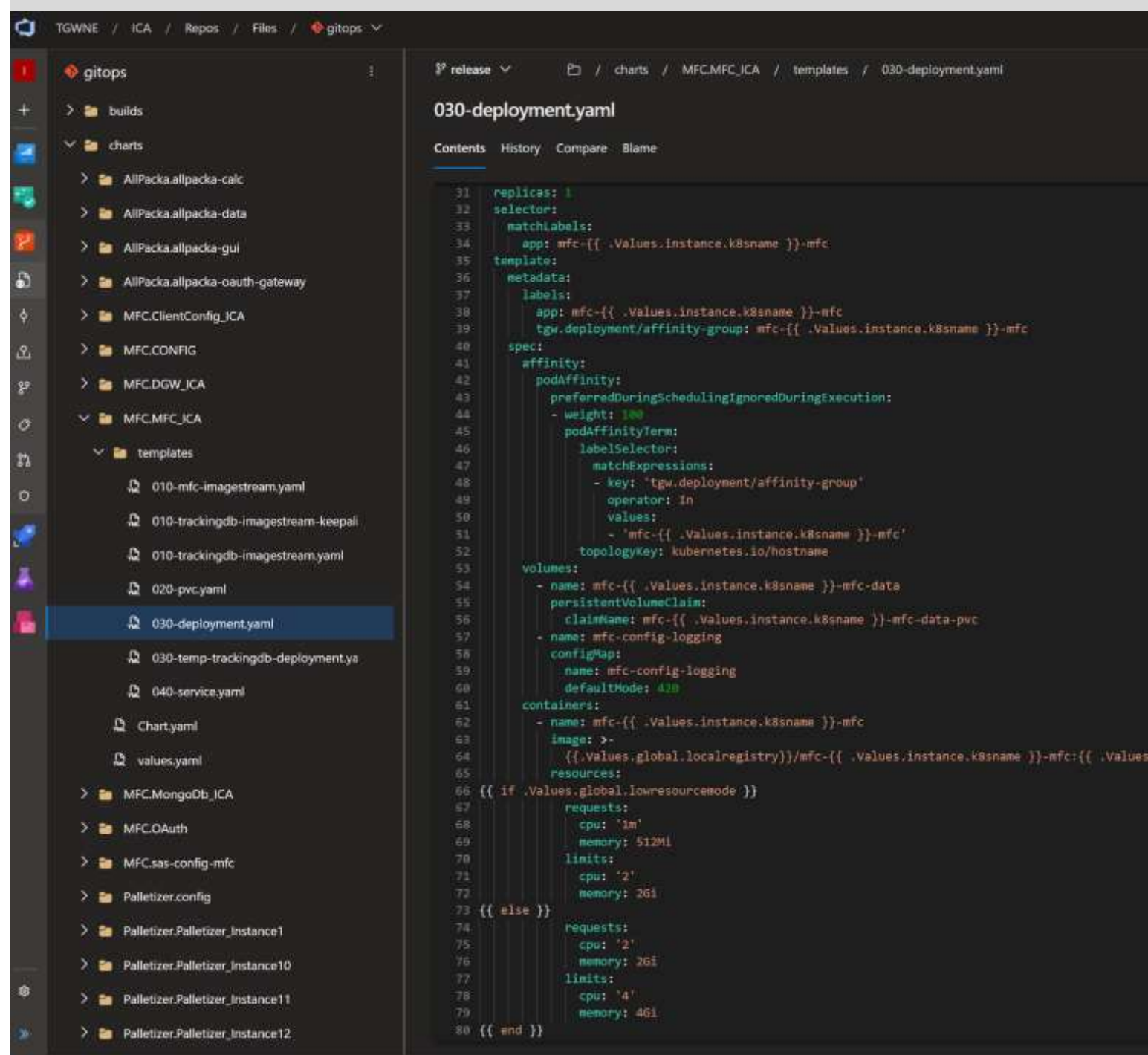
A single branch in the gitops repo can run in multiple namespaces in multiple clusters.



How we use openshift

A single branch in the gitops repo can run in multiple namespaces in multiple clusters.

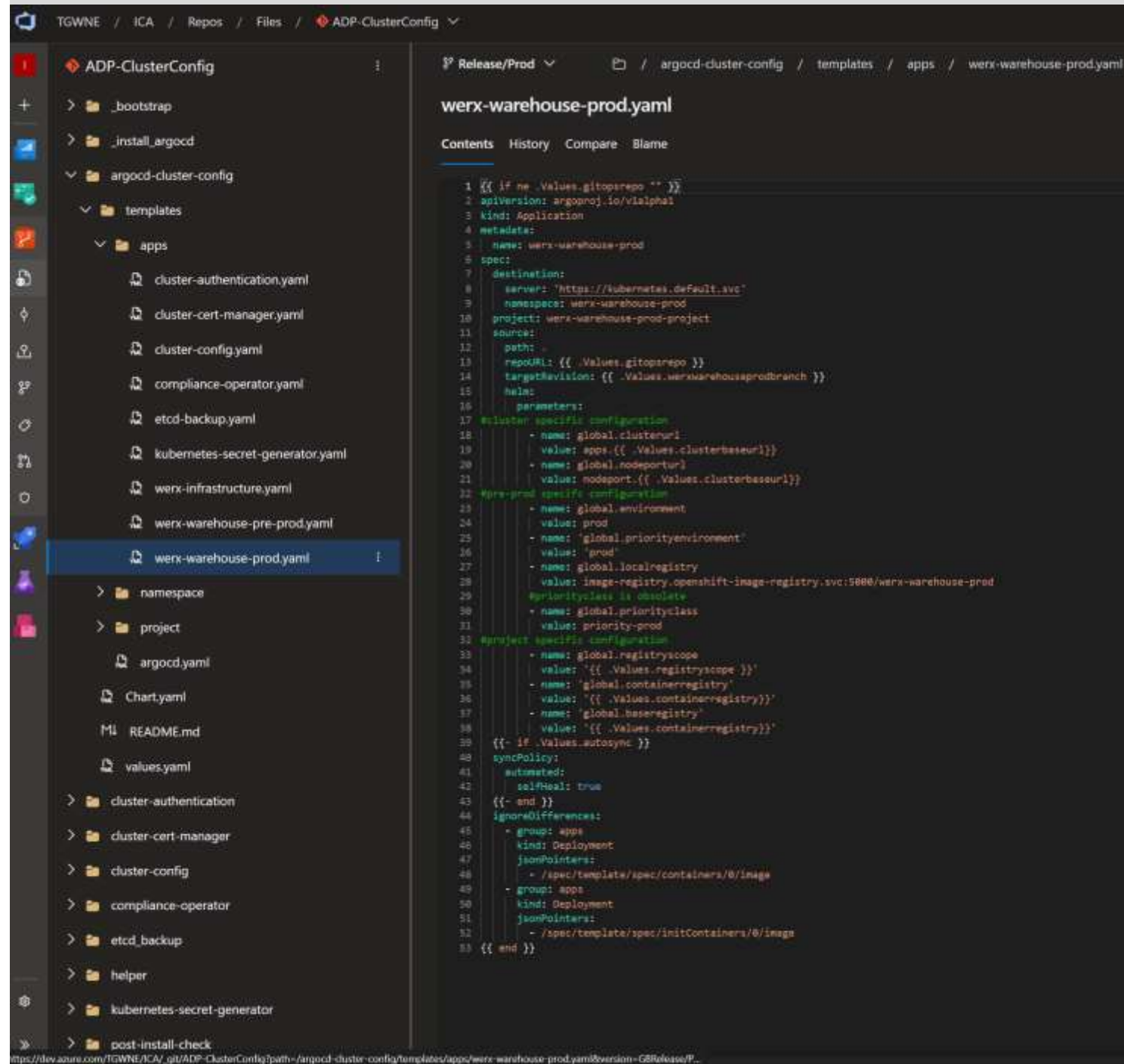
For this we're using lots of HELM variables in our kubernetes definitions that we're providing via argocd: cluster-url, prod/pre-prod/dev switch, ...



How we use openshift

In production environments those variables are passed through the argocd app-of-apps coming from the ADP-ClusterConfig repo.

(And the argocd app-of-apps gets some of the values from the installation script)



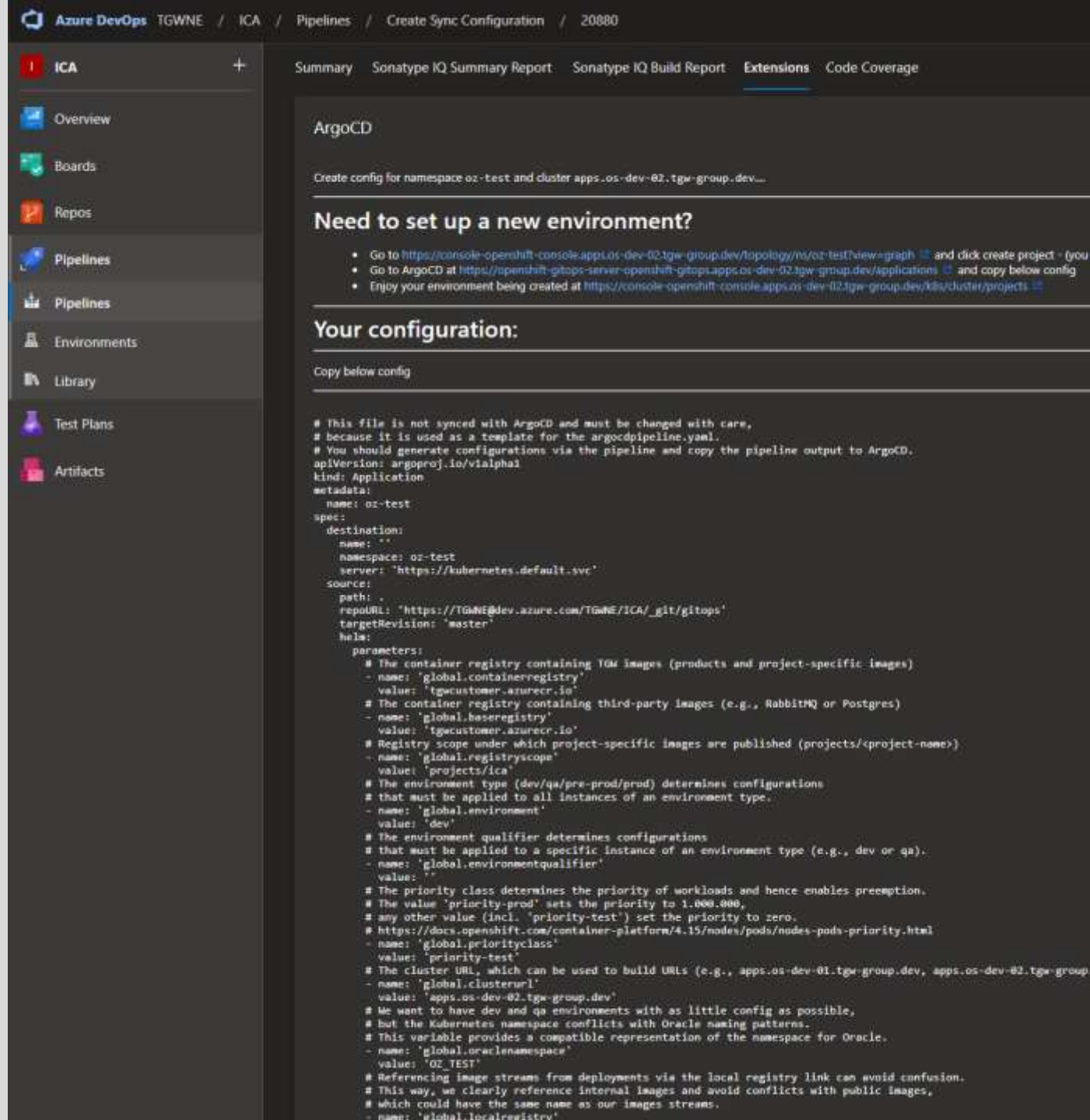
How we use openshift

Our internal test environments look very similar, but...

The app-of-apps and the cluster configs have HELM switches in it as well. If we have the inhouse flag, argocd allows self-service – developers can provision their environments themselves*.

To set this up each project has a pipeline. Fill in the target namespace you want to set up and you'll get the argocd configuration.

*At the moment manually – as we're dealing with lots of different organizations.



The screenshot shows the Azure DevOps interface for a pipeline named 'ICA'. The left sidebar contains navigation links for Overview, Boards, Repos, Pipelines, Environments, Library, Test Plans, and Artifacts. The main area displays the 'Summary' tab for the 'ICA' pipeline, which is configured to create a new environment. The 'Need to set up a new environment?' section provides instructions on how to create a project in ArgoCD. The 'Your configuration:' section shows a YAML configuration for the ArgoCD pipeline, which is a Helm chart for the 'openshift' namespace. The configuration includes parameters for the container registry, the environment type, the environment qualifier, the priority class, the cluster URL, and the namespace. The configuration is a Helm chart for the 'openshift' namespace, which is used to create a new environment. The configuration includes parameters for the container registry, the environment type, the environment qualifier, the priority class, the cluster URL, and the namespace. The configuration is a Helm chart for the 'openshift' namespace, which is used to create a new environment.

```
# This file is not synced with ArgoCD and must be changed with care,
# because it is used as a template for the argocdpipeline.yaml.
# You should generate configurations via the pipeline and copy the pipeline output to ArgoCD.
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: oz-test
spec:
  destination:
    name: ""
    namespace: oz-test
    server: "https://kubernetes.default.svc"
  source:
    path: .
    repoURL: "https://TGWNE@dev.azure.com/TGWNE/ICA/_git/gitops"
    targetRevision: "master"
  helm:
    parameters:
      - name: 'global.containerregistry'
        value: 'tgcustomer.azurecr.io'
      - name: 'global.baseregistry'
        value: 'tgcustomer.azurecr.io'
      - name: 'global.registryscope'
        value: 'projects/ica'
      - name: 'global.environment'
        value: 'dev'
      - name: 'global.environmentqualifier'
        value: ''
      - name: 'global.priorityclass'
        value: 'priority-test'
      - name: 'global.clusterurl'
        value: 'apps.os-dev-01.tgw-group.dev, apps.os-dev-02.tgw-group'
      - name: 'global.oraclenamespace'
        value: 'OZ_TEST'
      - name: 'global.localregistry'
        value: ''
```

How we use openshift

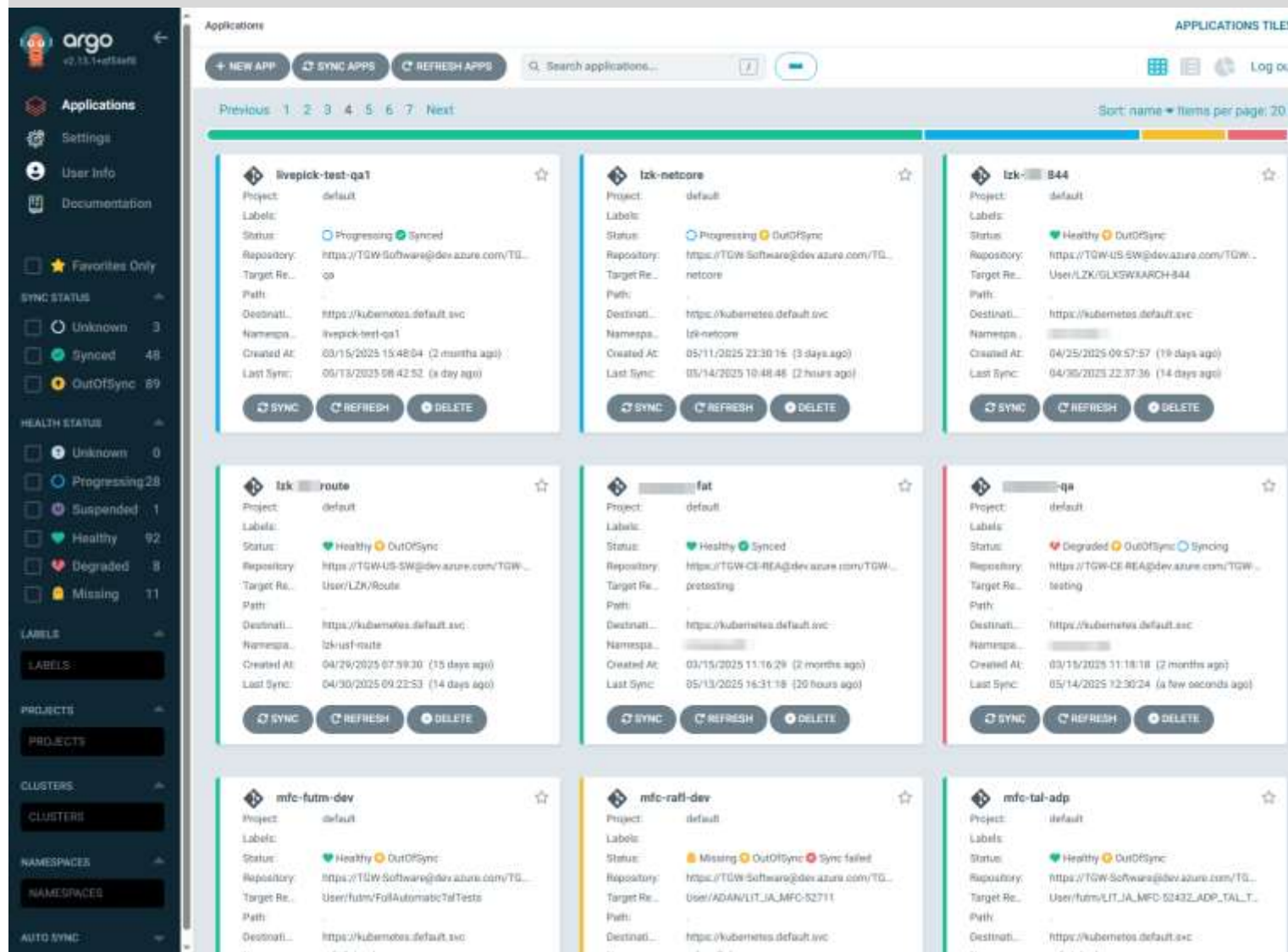
Internally we have 2 main clusters:

dev for regular day to day testing of new features. Around 120-140 full environments configured; spread out over around 25 nodes (usually 8 CPU, 32GB RAM) in multiple datacenter.

Only ~40 able to run concurrently.

On the one hand: CSI driver limits (60/node)

On the other hand: resource constraints *for development* are hard to estimate / set. CPU is never a problem. RAM is hard. Basic functional tests are OK. But as soon as some load goes into it, eviction errors cascade.



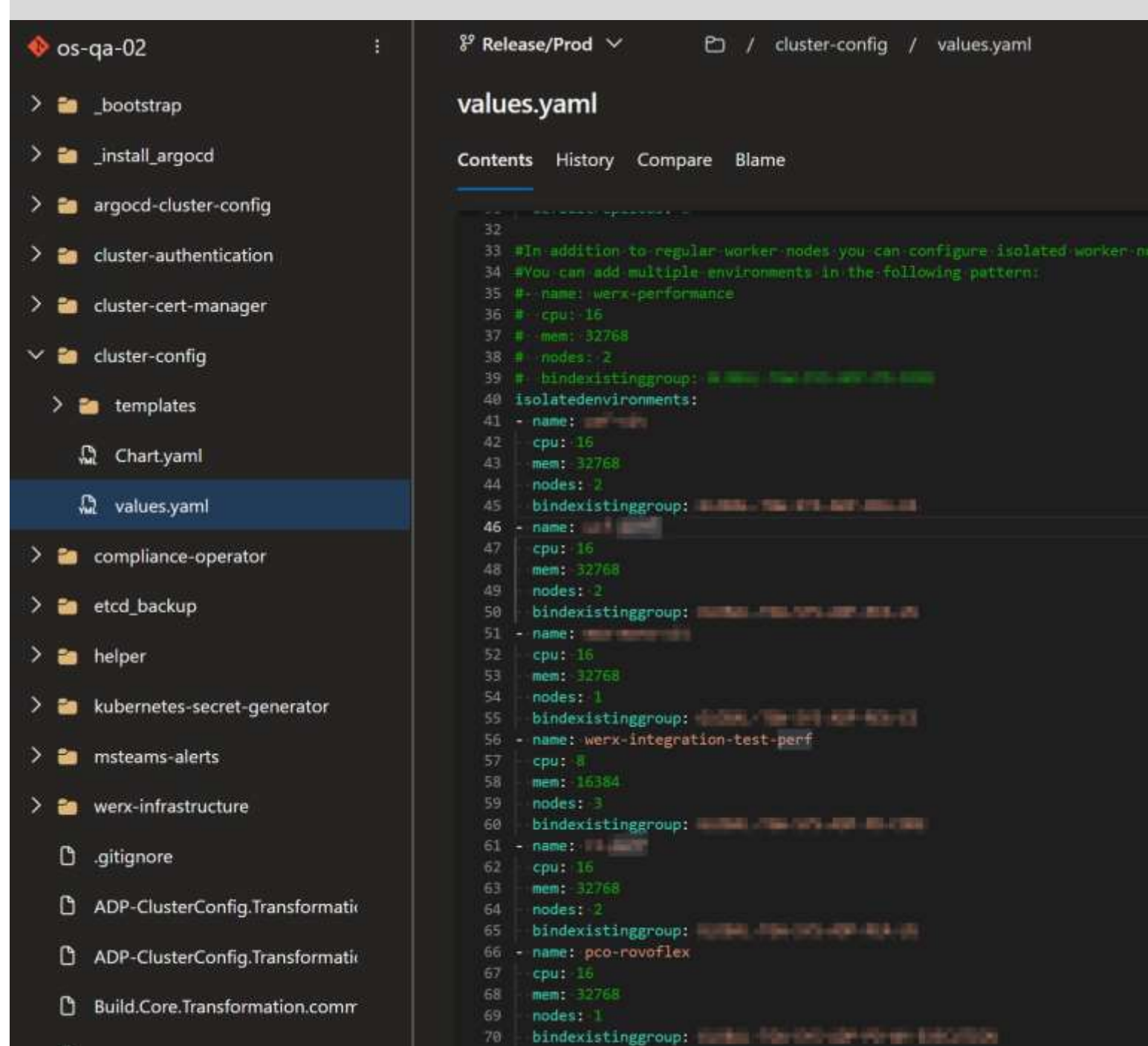
How we use openshift

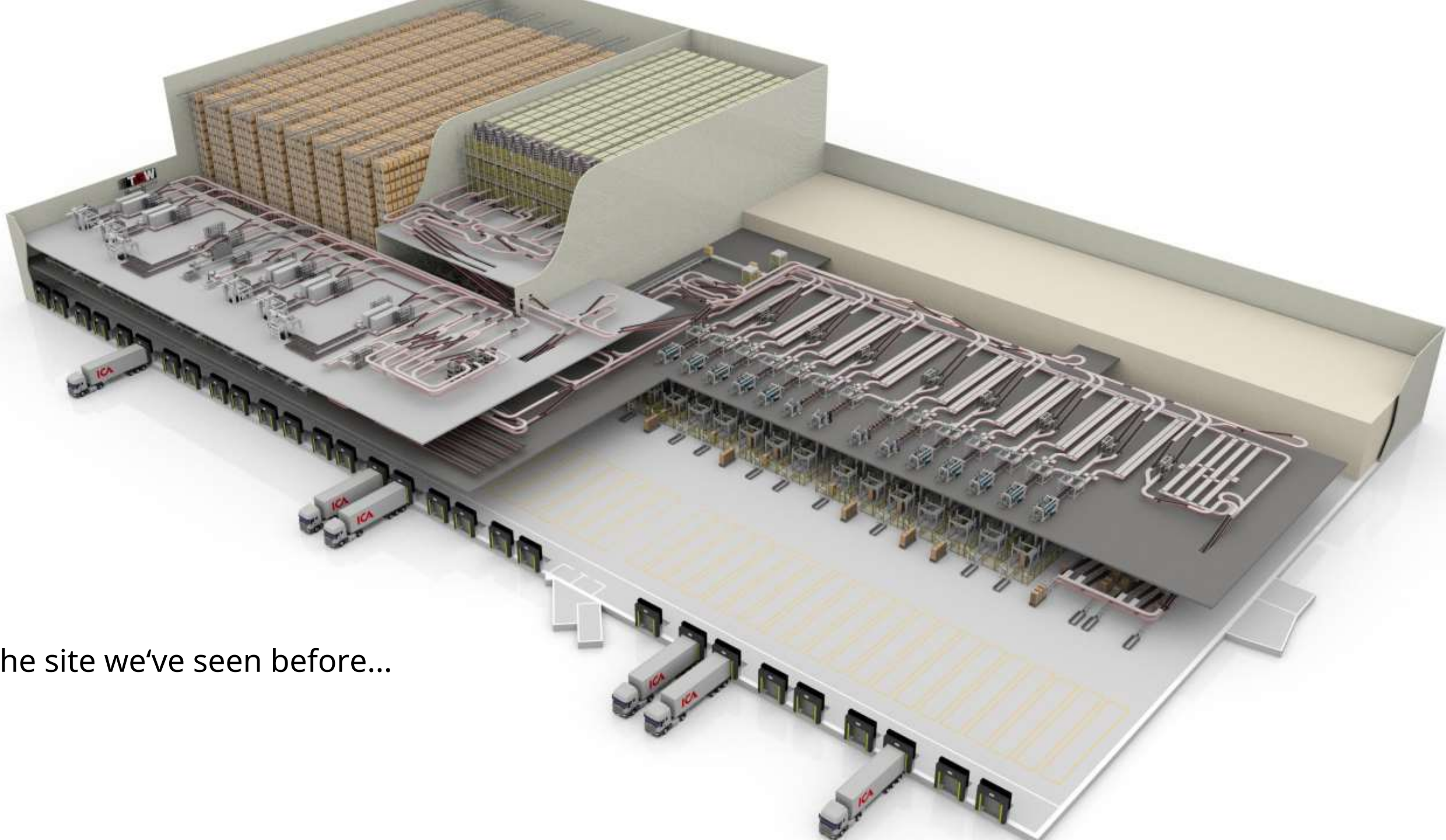
Internally we have 2 main clusters:

qa for special load tests where you need to have a „controlled“ environment that is comparable

(is my application now slower because of a bug or because there's additional load on the cluster).

Each namespace goes to dedicated nodes.





The site we've seen before...

Developer

+Add

Topology

Observe

Search

Builds

Helm

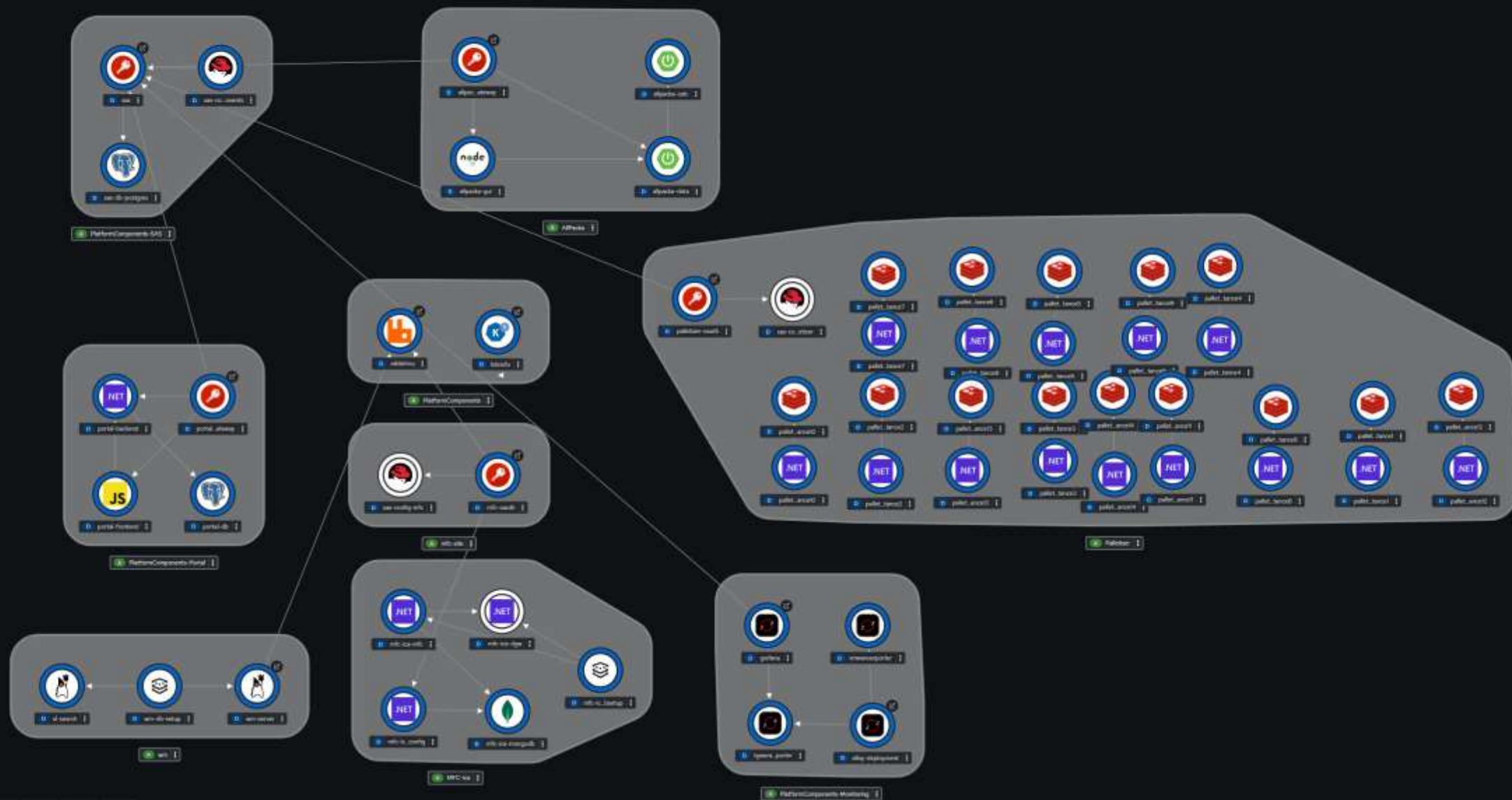
Project

ConfigMaps

Secrets

Project: ica-paro2 Application: All applications

Display options Filter by resource Name Find by name



- Developer
- +Add
- Topology
- Observe
- Search
- Builds
- Helm
- Project
- ConfigMaps
- Secrets

Project: ica-paro2

PR ica-paro2 Active

Overview Details

Details

[View all](#)

Name

ica-paro2

Requester

Leonard.Eames@tgw-group.com

Labels

kubernetes.io/metadata.name=ica-paro2

pod-security.kubernetes.io/audit=restricted

pod-security.kubernetes.io/audit-version=latest

[View all](#)

Description

No description

Inventory

57 Deployments

0 DeploymentConfigs

0 StatefulSets

56 Pods

43 PersistentVolumeClaims

59 Services

15 Routes

103 ConfigMaps

0 VolumeSnapshots

Status

Active



Alerts could not be loaded.

Utilization

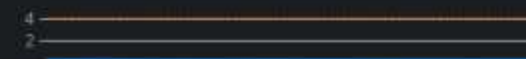
1 hour

Resource

Usage

CPU

292.5m



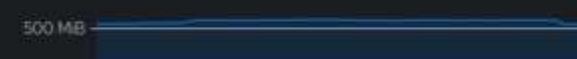
Memory

9.87 GiB



Filesystem

568.5 MiB



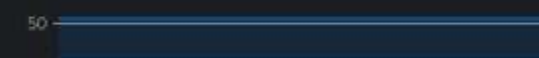
Network transfer

47.13 KBps in
40.47 KBps out



Pod count

56



ResourceQuotas

No ResourceQuotas

AppliedClusterResourceQuotas

No AppliedClusterResourceQuotas

Activity

Ongoing

There are no ongoing activities.

Recent events

There are no recent events.

So – how does all of this improve the initial setup?



Change

Before we started, setting up a new inhouse environment took a week of skilled effort.

Now it takes 10minutes:
1 minute of work, 9 minutes of waiting

(which is awesome the first few times you see deployment after deployment popping up in the Topology view)

Environments can be started/stopped on demand – as it can be automated entirely.





challenges

after 3 years
inhouse

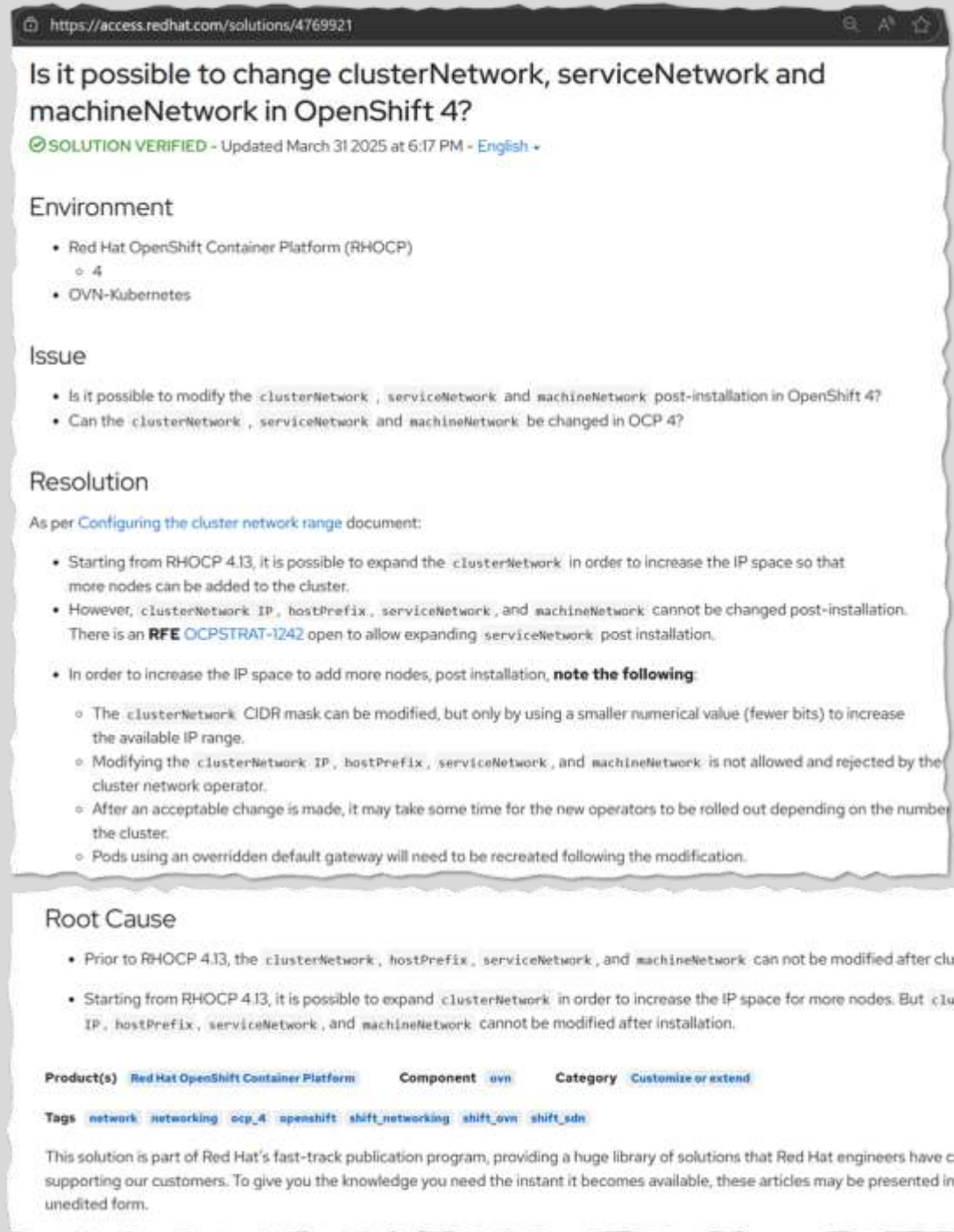
after 1 year at
customer

Challenges Network

When we install in a new customer warehouse, it is common that network is set up just prior.

Customer regularly wants to update network configuration (new IP addresses for master/ingress/api, change from DHCP to static or vice versa).

This in itself is no unreasonable request. However it's one that we usually push off as risks are too high.



The screenshot shows a web page from Red Hat's solution center. The URL is <https://access.redhat.com/solutions/4769921>. The title is "Is it possible to change clusterNetwork, serviceNetwork and machineNetwork in OpenShift 4?". It is marked as "SOLUTION VERIFIED" and was updated on March 31, 2025, at 6:17 PM. The article is in English.

Environment

- Red Hat OpenShift Container Platform (RHOCP)
 - 4
- OVN-Kubernetes

Issue

- Is it possible to modify the `clusterNetwork`, `serviceNetwork` and `machineNetwork` post-installation in OpenShift 4?
- Can the `clusterNetwork`, `serviceNetwork` and `machineNetwork` be changed in OCP 4?

Resolution

As per [Configuring the cluster network range](#) document:

- Starting from RHOCP 4.13, it is possible to expand the `clusterNetwork` in order to increase the IP space so that more nodes can be added to the cluster.
- However, `clusterNetwork IP`, `hostPrefix`, `serviceNetwork`, and `machineNetwork` cannot be changed post-installation. There is an [RFE OCPSTRAT-1242](#) open to allow expanding `serviceNetwork` post installation.
- In order to increase the IP space to add more nodes, post installation, **note the following**:
 - The `clusterNetwork` CIDR mask can be modified, but only by using a smaller numerical value (fewer bits) to increase the available IP range.
 - Modifying the `clusterNetwork IP`, `hostPrefix`, `serviceNetwork`, and `machineNetwork` is not allowed and rejected by the cluster network operator.
 - After an acceptable change is made, it may take some time for the new operators to be rolled out depending on the number the cluster.
 - Pods using an overridden default gateway will need to be recreated following the modification.

Root Cause

- Prior to RHOCP 4.13, the `clusterNetwork`, `hostPrefix`, `serviceNetwork`, and `machineNetwork` can not be modified after cluster installation.
- Starting from RHOCP 4.13, it is possible to expand `clusterNetwork` in order to increase the IP space for more nodes. But `clusterNetwork IP`, `hostPrefix`, `serviceNetwork`, and `machineNetwork` cannot be modified after installation.

Product(s) [Red Hat OpenShift Container Platform](#) **Component** [ovn](#) **Category** [Customize or extend](#)

Tags [network](#) [networking](#) [ocp_4](#) [openshift](#) [shift_networking](#) [shift_ovn](#) [shift_sdn](#)

This solution is part of Red Hat's fast-track publication program, providing a huge library of solutions that Red Hat engineers have created while supporting our customers. To give you the knowledge you need the instant it becomes available, these articles may be presented in a raw and unedited form.

Challenges Network

Bites us internally as well...

The current dev cluster is a year old and should move to different servers.
Actual migration to servers is no problem (happens on VM Layer) – but this server has policy to use different VLANs.

No one dares to migrate there, so we need to install a new dev system, migrate applications, kill the old one.

Not rocket science, no live migration needed. But – effort for 100 people.

The screenshot shows a web page from Red Hat's OpenShift solutions library. The URL is <https://access.redhat.com/solutions/4769921>. The article title is "Is it possible to change clusterNetwork, serviceNetwork and machineNetwork in OpenShift 4?". It is marked as a "SOLUTION VERIFIED" and was updated on March 31, 2025, at 6:17 PM in English. The article is categorized under "Environment" (Red Hat OpenShift Container Platform (RHOCP) 4, OVN-Kubernetes) and "Issue" (Is it possible to modify the clusterNetwork, serviceNetwork and machineNetwork post-installation in OpenShift 4?; Can the clusterNetwork, serviceNetwork and machineNetwork be changed in OCP 4?). The "Resolution" section refers to the "Configuring the cluster network range" document and lists several points: starting from RHOCP 4.13, it's possible to expand the clusterNetwork; however, clusterNetwork IP, hostPrefix, serviceNetwork, and machineNetwork cannot be changed post-installation; there is an RFE OCPSTRAT-1242 for expanding serviceNetwork; and to increase IP space, one must note that the clusterNetwork CIDR mask can be modified (but only with a smaller numerical value), modifying the clusterNetwork IP, hostPrefix, serviceNetwork, and machineNetwork is not allowed, and it may take time for new operators to be rolled out. The "Root Cause" section states that prior to RHOCP 4.13, these network parameters could not be modified after installation, but starting from 4.13, they can be expanded. The article is tagged with "network", "networking", "ocp_4", "openshift", "shift_networking", "shift_ovn", and "shift_sdn". It is part of Red Hat's fast-track publication program.

<https://access.redhat.com/solutions/4769921>

Is it possible to change clusterNetwork, serviceNetwork and machineNetwork in OpenShift 4?

SOLUTION VERIFIED - Updated March 31 2025 at 6:17 PM - [English](#)

Environment

- Red Hat OpenShift Container Platform (RHOCP)
 - 4
- OVN-Kubernetes

Issue

- Is it possible to modify the clusterNetwork, serviceNetwork and machineNetwork post-installation in OpenShift 4?
- Can the clusterNetwork, serviceNetwork and machineNetwork be changed in OCP 4?

Resolution

As per [Configuring the cluster network range](#) document:

- Starting from RHOCP 4.13, it is possible to expand the clusterNetwork in order to increase the IP space so that more nodes can be added to the cluster.
- However, clusterNetwork IP, hostPrefix, serviceNetwork, and machineNetwork cannot be changed post-installation. There is an [RFE OCPSTRAT-1242](#) open to allow expanding serviceNetwork post installation.
- In order to increase the IP space to add more nodes, post installation, **note the following**:
 - The clusterNetwork CIDR mask can be modified, but only by using a smaller numerical value (fewer bits) to increase the available IP range.
 - Modifying the clusterNetwork IP, hostPrefix, serviceNetwork, and machineNetwork is not allowed and rejected by the cluster network operator.
 - After an acceptable change is made, it may take some time for the new operators to be rolled out depending on the number the cluster.
 - Pods using an overridden default gateway will need to be recreated following the modification.

Root Cause

- Prior to RHOCP 4.13, the clusterNetwork, hostPrefix, serviceNetwork, and machineNetwork can not be modified after cluster installation.
- Starting from RHOCP 4.13, it is possible to expand clusterNetwork in order to increase the IP space for more nodes. But clusterNetwork IP, hostPrefix, serviceNetwork, and machineNetwork cannot be modified after installation.

Product(s) [Red Hat OpenShift Container Platform](#) **Component** [ovn](#) **Category** [Customize or extend](#)

Tags [network](#) [networking](#) [ocp_4](#) [openshift](#) [shift_networking](#) [shift_ovn](#) [shift_sdn](#)

This solution is part of Red Hat's fast-track publication program, providing a huge library of solutions that Red Hat engineers have created while supporting our customers. To give you the knowledge you need the instant it becomes available, these articles may be presented in a raw and unedited form.

Challenges Storage



This is the biggest pain point for us at the moment – as it's the one we have least experience with:

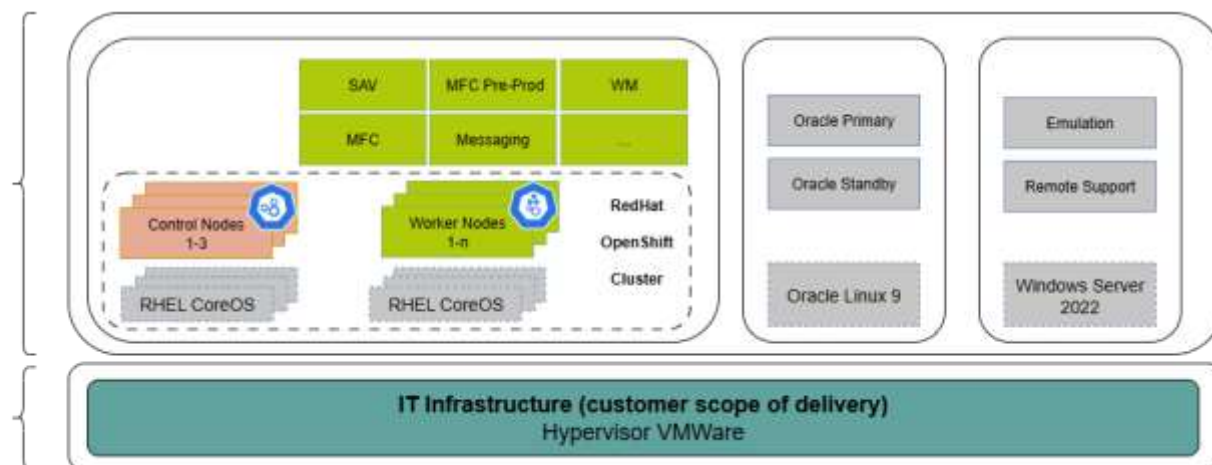
Historically, we either ran black box where we had control of everything.

Or we ran grey box where customer was in control of everything – and we got storage through the hypervisor layer.

Customer could provision disks for us in the hypervisor, but we were not required to know/deal/manage storage details.



Customer

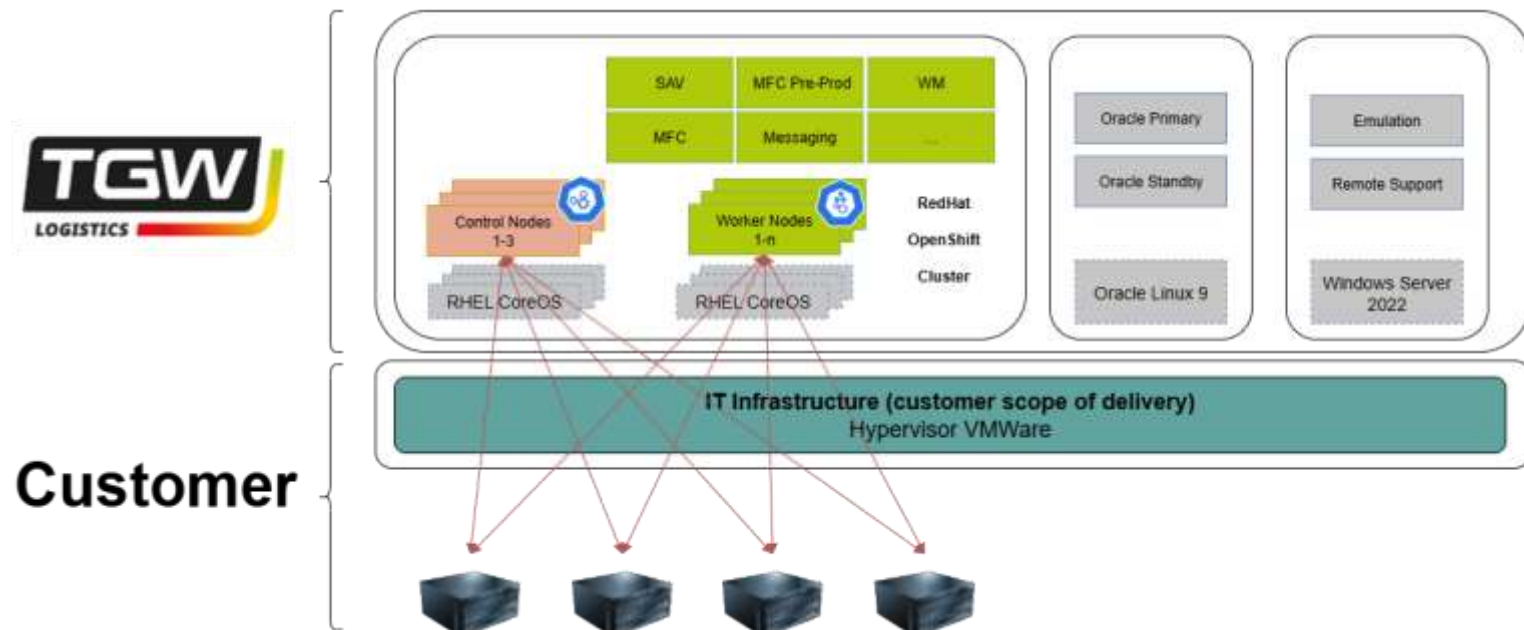


Challenges Storage

In grey box we're now running on top of the hypervisor – but need to interact to with storage/nodes below.

But...

Reluctant to give us storage mgmt access for CSI driver, not seeing the need for it altogether (just add storage to the VMs,...), much more complicated from security point of view

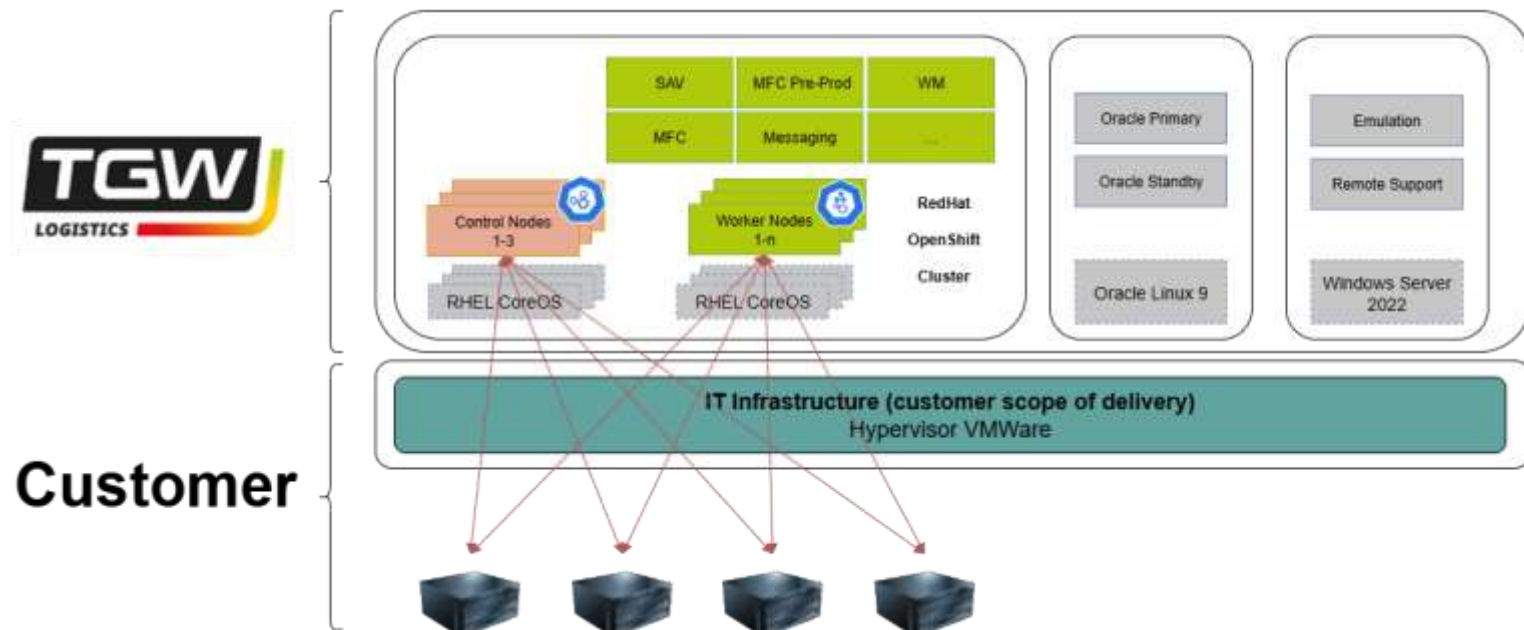


Challenges Storage

We were not addressing storage like rook/ceph (yet).

It definitely brings up lots of questions and uncertainties regarding support.
Shy away of complexity / subscription costs.

Part of it can be solved during pre-sales / contractual phase.



A futuristic industrial scene featuring a robotic arm with a red and white striped boom, positioned over a series of grey storage bins. In the background, a complex wireframe structure, possibly a building or a large machine, is visible. The scene is set against a dark, hazy background with a bright light source at the top, creating a high-tech, industrial atmosphere.

outlook

Outlook Arbiter Nodes

Many / most customers only provide two fault domains – which is not easy to fit 3 control nodes onto.

4.19 / 4.20 bring arbiter nodes that we plan to use: f.e. 2 nodes in DC, 1 arbiter node maybe outside of DC in OT Rack.

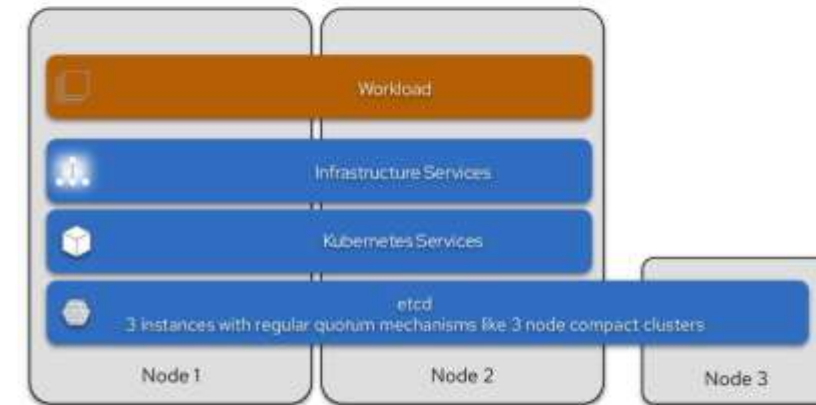
Two Node OpenShift with Arbiter (TNA)

Approach:

- Two node solution for cost sensitive customers
- Small arbiter node, running only 3d etcd instance
- Technically a three node cluster
- OCP Virtualization fully supported
- Hyperconverged Storage / SDS via Partners
- X86 and Arm, bare metal only

Timeline Targets:

- V4.19 Technology Preview
- V4.20 General Availability

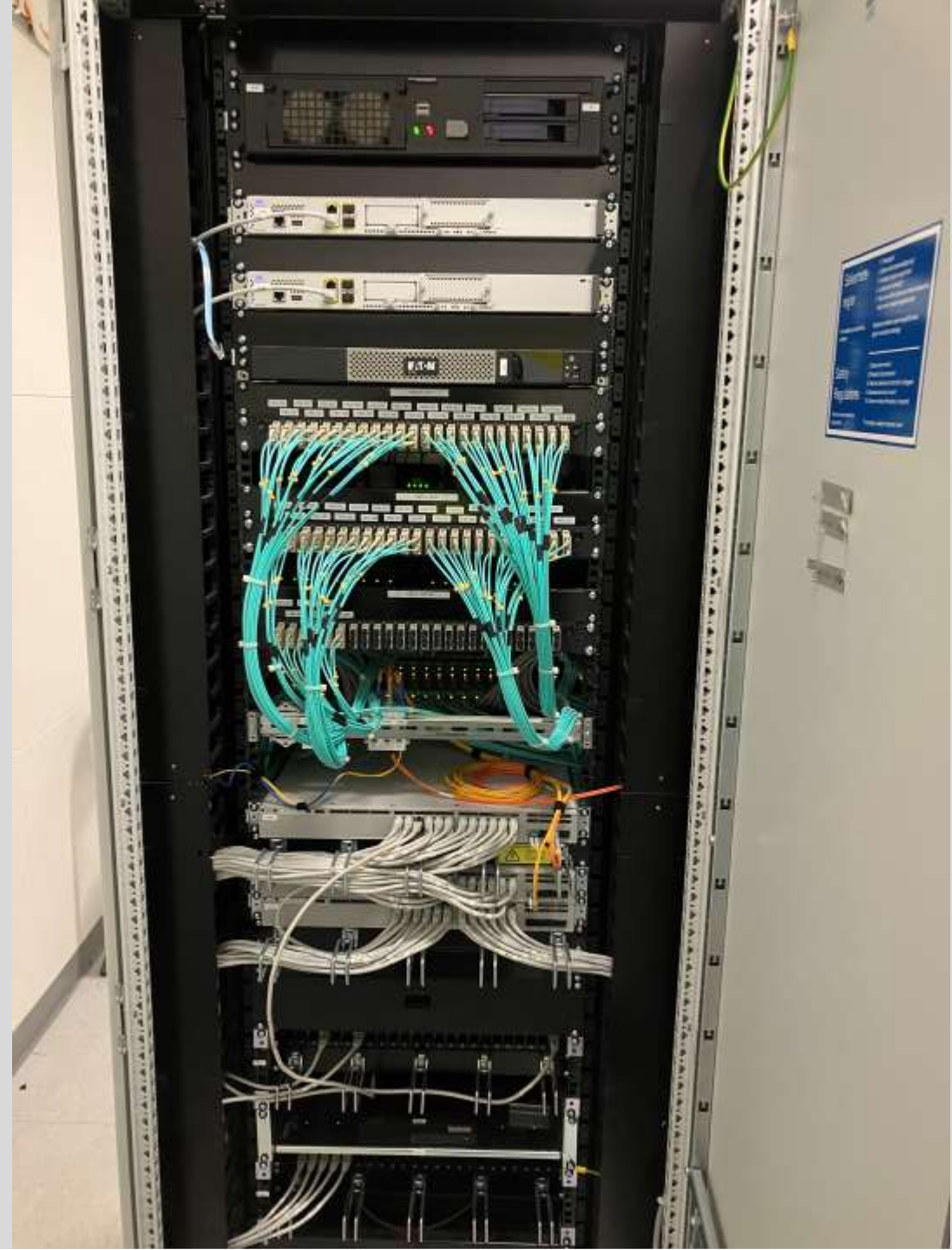


What's Next in OpenShift Q3 CY2025

Outlook Arbiter Nodes

Regardless of black / grey box we always ship OT Racks for device communication.

Although slower connectivity compared to DC (gigabit only) this may be an option for arbiter node.



Three large, overlapping, leaf-like shapes in red, yellow, and green. The yellow shape is in the center and contains the text "Thank you". The red shape is to the left, and the green shape is to the right. They all have a wavy, organic edge.

**Thank
you**