INSTANA
an IBM Company

SRE

Take the climb to MultiCloud

by Marcel Birkner

Bio

@MarcelBirkner

github.com/marcelbirkner

Marcel works as a Staff Site Reliability Engineer at Instana, an Application Performance Monitoring (APM) solution. He has long experience in software engineering and software automation. Currently he focuses on improving the current Kubernetes stack, reducing overall system complexity and installing Instana SaaS infrastructure in IBM Cloud.

INSTANA an IBM Company

Abstract

For Instana MultiCloud is not just a buzzword, but an opportunity to grow our customer base. We initially offered our SaaS solution in AWS Cloud. Last year we opened new SaaS regions in Google Cloud and this year we are adding SaaS regions in IBM Cloud.

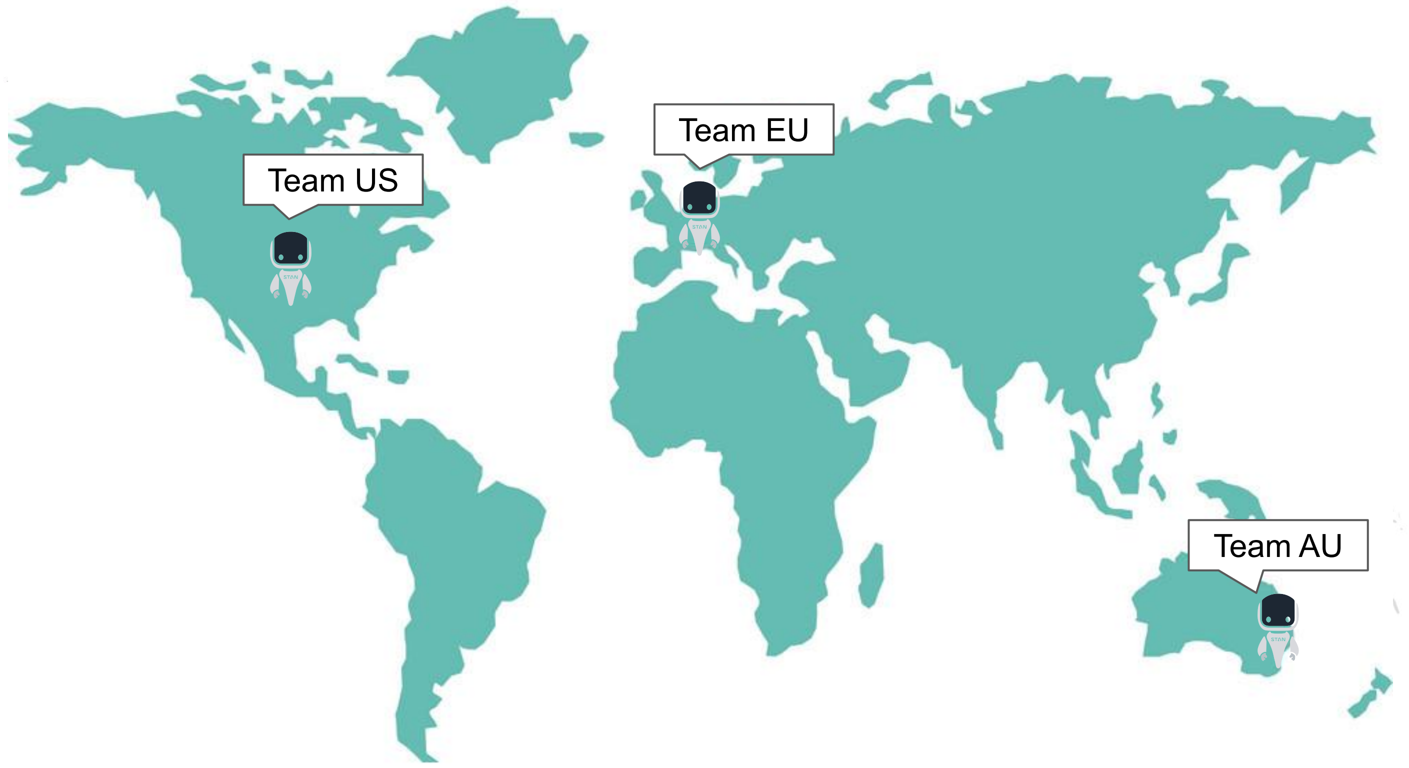
We knew that the platform and infrastructure that got us through the first five years needed an overhaul to prepare us for more growth. Our customers have strict requirements regarding compliance, security and data governance. That is when we decided to update our infrastructure to be able to open new SaaS regions with other cloud providers.

I will present the challenges we faced during the last two years. Running the old stack, not breaking existing customers and designing and implementing our new infrastructure that will serve us the next five years.

Who We Are

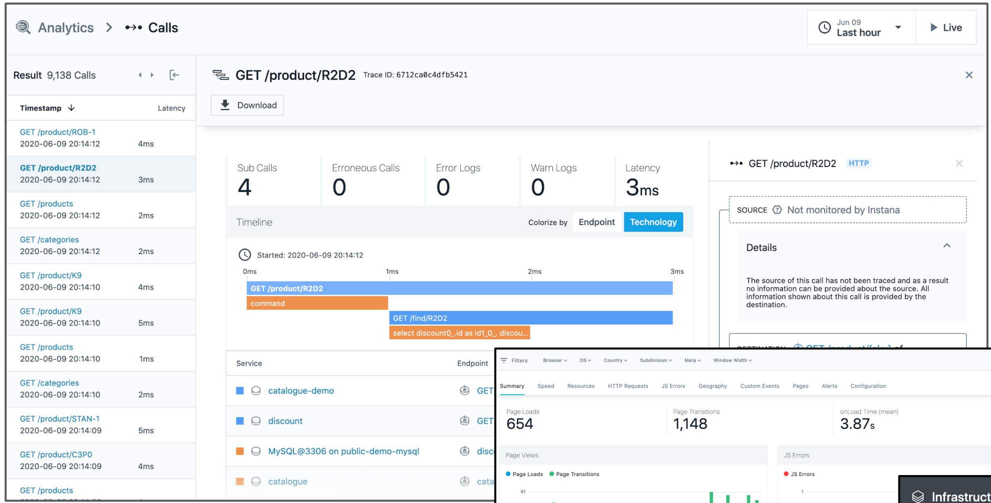
SRE Team

- 3 Time zones
- 24 / 7 / 365 support
- On-call rotation
- Team members have operations and software engineering background



INSTANA

What We Do



Analytics > Calls

Jun 09 Last hour | Live

Result 9,138 Calls

Timestamp	Latency
GET /product/ROB-1 2020-06-09 20:14:12	4ms
GET /product/R2D2 2020-06-09 20:14:12	3ms
GET /products 2020-06-09 20:14:12	2ms
GET /categories 2020-06-09 20:14:12	2ms
GET /product/K9 2020-06-09 20:14:10	4ms
GET /product/K9 2020-06-09 20:14:10	5ms
GET /products 2020-06-09 20:14:10	1ms
GET /categories 2020-06-09 20:14:10	2ms
GET /product/STAN-1 2020-06-09 20:14:09	5ms
GET /product/C3P0 2020-06-09 20:14:09	4ms
GET /products	

GET /product/R2D2 Trace ID: 6712ca0c4dfb5421

Download

Sub Calls	Erroneous Calls	Error Logs	Warn Logs	Latency
4	0	0	0	3ms

Timeline — Colorize by Endpoint | Technology

Started: 2020-06-09 20:14:12

0ms 1ms 2ms 3ms

GET /product/R2D2
command
GET /find/R2D2
select discount0_.id as id1_0_, discou...

Service	Endpoint
catalogue-demo	GET
discount	GET
MySQL@3306 on public-demo-mysql	disc
catalogue	cata

GET /product/R2D2 HTTP

SOURCE Not monitored by Instana

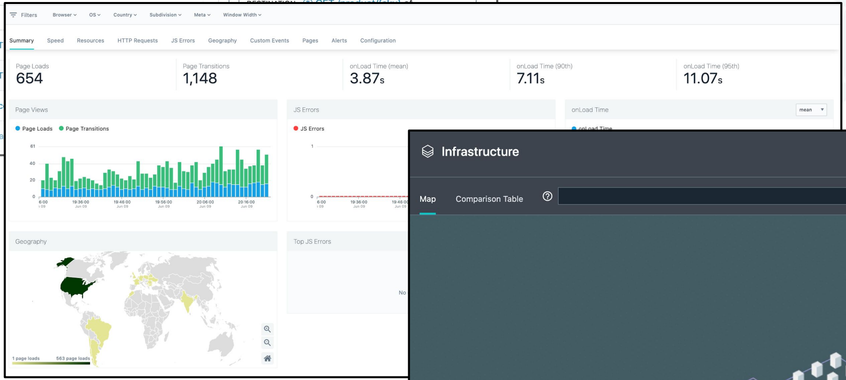
Details

The source of this call has not been traced and as a result no information can be provided about the source. All information shown about this call is provided by the destination.

Infrastructure

Service

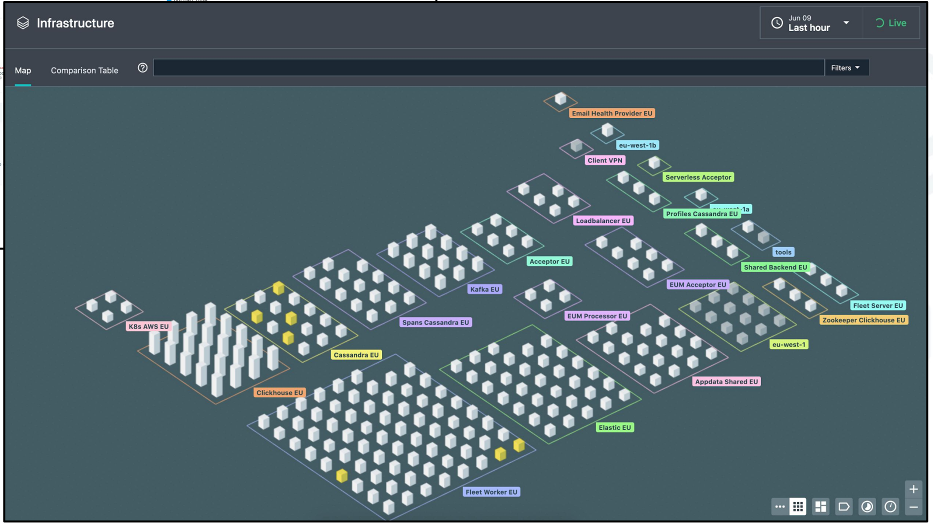
Application



Filters Browser OS Country Subdivision Meta Window Width

Summary Speed Resources HTTP Requests JS Errors Geography Custom Events Pages Alerts Configuration

Page Loads	Page Transitions	onLoad Time (mean)	onLoad Time (90th)	onLoad Time (95th)
654	1,148	3.87s	7.11s	11.07s



Infrastructure

Jun 09 Last hour | Live

Map Comparison Table Filters

Tracing

EUM

Mobile App

https://play-with.instana.io

Stats

- 10 different datastore clusters per region
- 8K+ Containers Running in SaaS

1M+
TRACES/SEC ANALYZED

300K+
CONTAINERS MONITORED

140K+
JVMS MONITORED

60K+
PODS MONITORED

50K+
HOSTS MONITORED

20K+
FUNCTIONS MONITORED

SaaS stats from 2020

Our MultiCloud Journey

Where we were 2018

SAAS:

- Single Cloud Provider
- 2 x AWS regions
- HashiCorp (Nomad/Consul)
- Ansible playbooks

On-Premises:

- package based
- Chef cookbooks

Challenges

- **Growth**
- **Compliance/Security/Data governance**

INSTANA

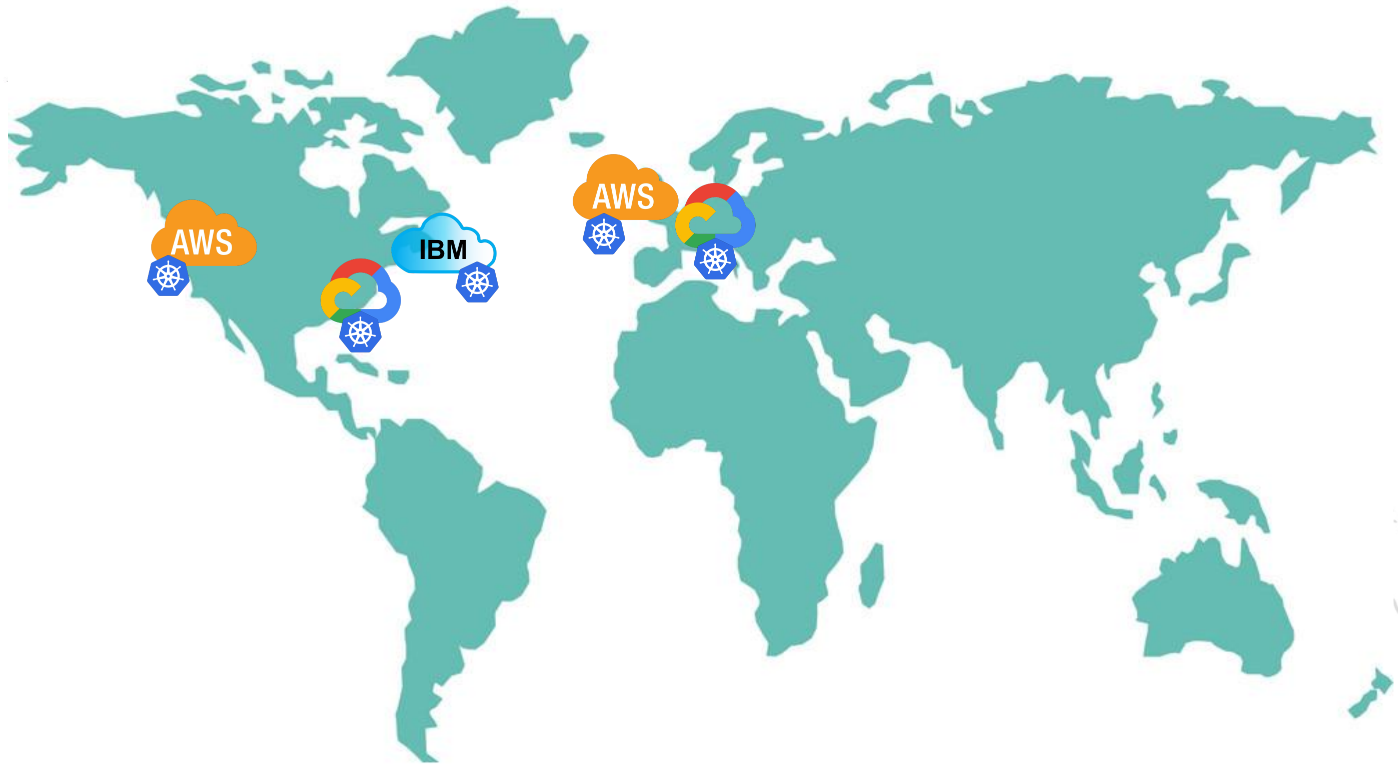
2021

SaaS:

- Multi Cloud Strategy
- 2 x AWS regions
- 2 x GCP regions
- first IBM region (internal customers only atm)
- Kubernetes

On-Premises:

- Docker
- Kubernetes



INSTANA

Identify Challenges

Identify challenges

- What is **working well** in the current infrastructure?
- What needs to be **improved**?
- How can we save **daily toil**?
- How do we want to run SaaS product in the **future**?

Focus on the big picture

- try not to solve all problems at once
- some requirements will change

Our "Big Picture"

- Kubernetes
- Shared configuration / code for SaaS and On-Premises
- Reduce complexity / toil

Goal #1: Single datastore migration codebase (SaaS / On-Premises)

up to 2019

Challenges: Each datastore had its own migration tool. Duplicate scripts for SaaS and OnPrem.

- Cassandra (cassandra-migrator)
- ClickHouse (golang-migrate)
- Elasticsearch (http-client)
- Kafka (kafka-cli)
- MongoDB (mongo migrator)
  - replaced by CockroachDB
- PostgreSQL (flyway db)
  - replaced by CockroachDB

Runtimes: Ruby/Python/Java

2020

instanactl

- GoLang CLI
  - cobra library
  - golang-migrate library
- codebase used by SaaS and On-Premises
- single place for database migration scripts

Runtimes: GoLang Binary

Goal #2: Shared configuration & codebase (SaaS / On-Premises)

up to 2019

Challenges:

- separate component configuration
- separate packaging
  - SaaS: Docker
  - OnPrem: RPM / DEB
- separate delivery
  - SaaS: Ansible
  - OnPrem: Chef

Runtimes: Python / Ruby

2020

- shared component configuration
- shared OCI container images
- shared migration tool
- K8s deployments via **instanactl**

Runtimes: GoLang Binary

Supported Operating Systems

Ubuntu, Debian, RedHat, CentOS, Amazon Linux

Goal #3: Infra. config versioned with product (SaaS / On-Premises)

up to 2019

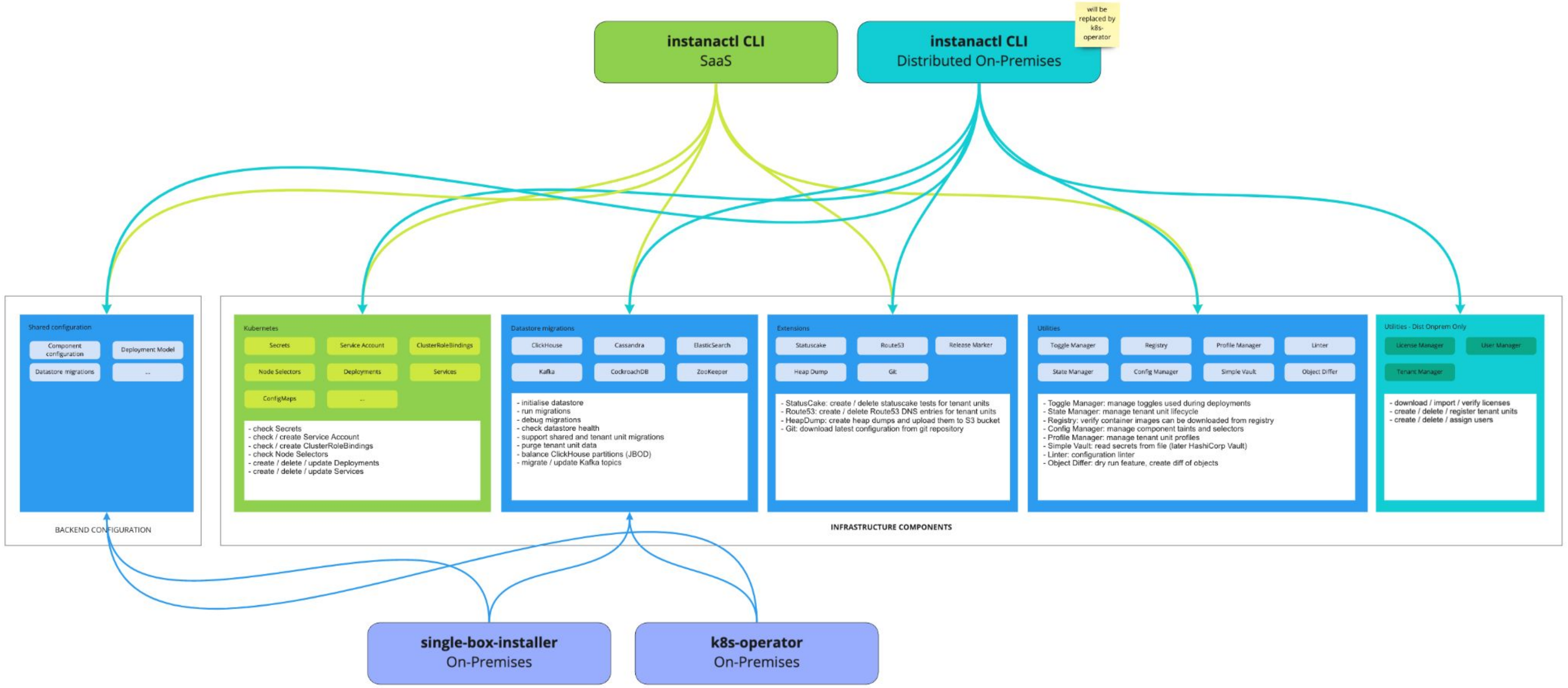
Challenges:

- SaaS and OnPrem had separate repositories for datastore migrations and component configuration
- no common versioning with product source code
- hard to coordinate releases and hotfixes

2020

- Mono-Repo for product source code, component configuration and datastore migration scripts
    - release branches (release-199, release-200, ...)
- releases are easily rolled out from release branches
- easy coordination of SaaS and OnPrem releases and hotfixes

Shared Infrastructure Modules (SaaS / On-Premises)



- check Secrets
- check / create Service Account
- check / create ClusterRoleBindings
- check Node Selectors
- create / delete / update Deployments
- create / delete / update Services

- initialise datastore
- run migrations
- debug migrations
- check datastore health
- support shared and tenant unit migrations
- purge tenant unit data
- balance ClickHouse partitions (JBOD)
- migrate / update Kafka topics

- StatusCake: create / delete statuscake tests for tenant units
- Route53: create / delete Route53 DNS entries for tenant units
- HeapDump: create heap dumps and upload them to S3 bucket
- Git: download latest configuration from git repository

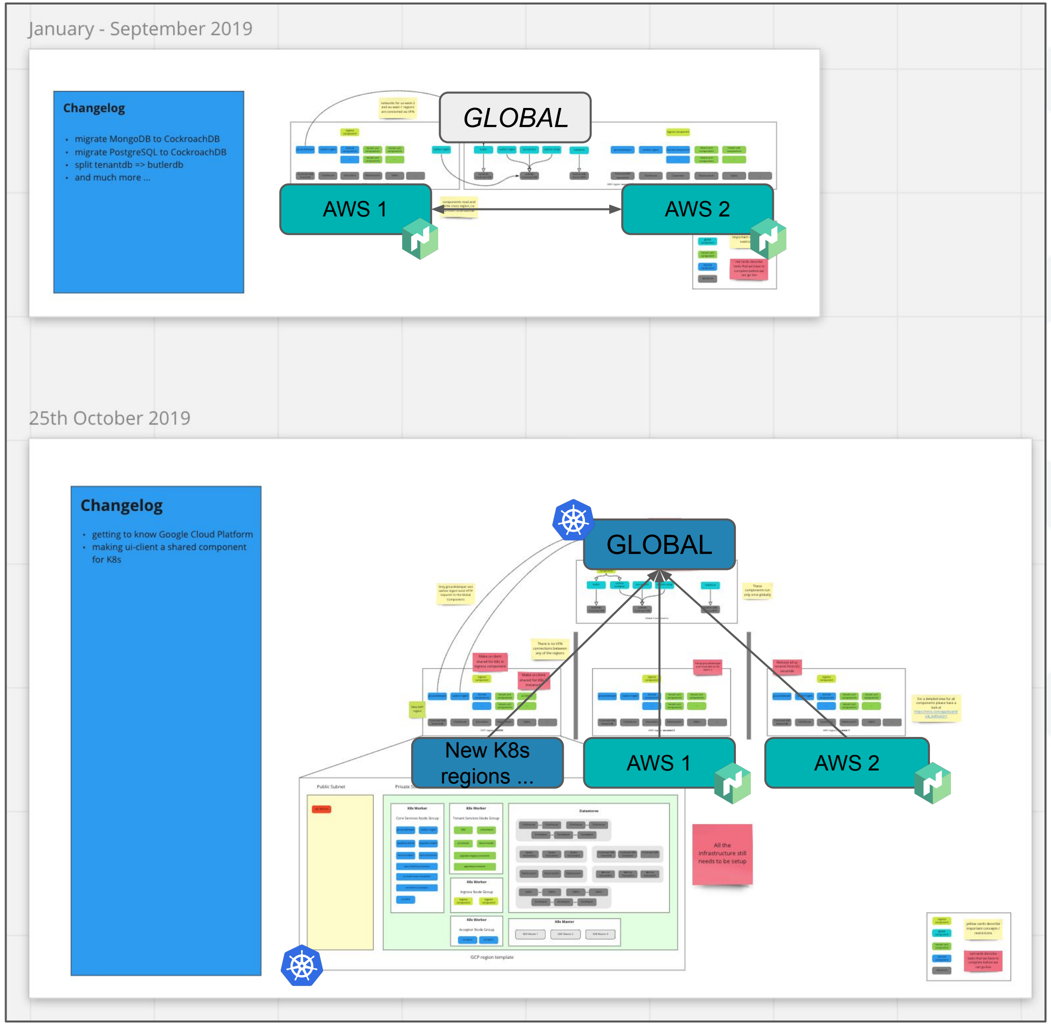
- Toggle Manager: manage toggles used during deployments
- State Manager: manage tenant unit lifecycle
- Registry: verify container images can be downloaded from registry
- Config Manager: manage component taints and selectors
- Profile Manager: manage tenant unit profiles
- Simple Vault: read secrets from file (later HashiCorp Vault)
- Linter: configuration linter
- Object Differ: dry run feature, create diff of objects

- download / import / verify licenses
- create / delete / register tenant units
- create / delete / assign users

Migration process

Infrastructure & Code Changes

- new architecture (global vs regional components)
- migrate MongoDB + PostgreSQL to CockroachDB
- component refactoring



January - September 2019

Changelog

- migrate MongoDB to CockroachDB
- migrate PostgreSQL to CockroachDB
- split tenantdb => butlerdb
- and much more ...

25th October 2019

Changelog

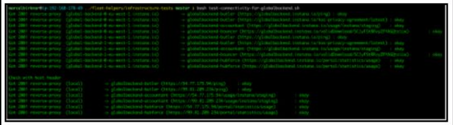
- getting to know Google Cloud Platform
- making ui-client a shared component for K8s

INSTANA an IBM Company

"GLOBAL" environment GoLive Steps

This document contains all steps for taking the new K8s rainbow environment live. Go live steps need to be executed in order.
Each Go Live step contains the name of a person that is responsible for verifying that the feature is working.

Preparation

ID	Task	Responsible	Status
P1	Prepare PR with production values for **rainbow.hcl (secrets, hubspot, oauth, ...)** • compare config files with production • https://github.com/instana/instanactl/tree/release-172/scripts/config Config change: https://github.com/instana/instanactl/commit/7e4fc91cc4218d7f92b004a274d77b5bb968aa0a	MB / CS	done
P2	Verify that production deployment jobs are working for **rainbow** • https://orchestration-rainbow.instana.io/job/instanactl-update-global/	Marcel	done
P3	Test hubforce migrations are working against rainbow • https://orchestration-rainbow.instana.io/view/hubforce/job/hubforce-run-db-migrator/	Schmitzi, Vedran	done
P4	Prepare PR with production values for **hubforce (secrets, aurora, datastores, sqs config, ...)** • compare config files with production • https://github.com/instana/hubforce/blob/develop/hubforce-core/deployment-k8s/configmap/rainbow.yml.enc	Schmitzi, Vedran	done
P5	Verify that production deployment jobs are working for hubforce • https://orchestration-rainbow.instana.io/job/hubforce-deploy/	Schmitzi, Vedran	done
P6	Scaleout **cashier-acceptor & cashier-ingest** to match production (see pod listing below)	MB / CS	done
P7	Run infrastructure network tests before rollout • https://github.com/instana/fleet-helpers/tree/master/infrastructure-tests 	MB / CS	done
P8	Rollout latest instanactl version to https://orchestration-rainbow.instana.io	Marcel	done
P9	Check certificates in K8s rainbow (instana.io)	MB / CS	done
P10	Check K8s cluster / worker / security groups	MB / CS	done
P11	Build latest K8s images from release-172 (2.172.175-0)	MB / CS	done
P20	Check rainbow components sizing / profiles (compare with SAAS) • changed to profile "xlarge" for global components	MB / CS	done
P21	Prepare config PR for global-backend nginx (reverse proxy) to forward requests to K8s rainbow • create backup of existing nginx config on server for rollback • https://github.com/instana/cookbooks/blob/master/fleet-loadbalancer/templates/default/global-backend.erb	MB / CS	done
P22	Check "butler" Consul entry in EU (to point to reverse proxy) https://github.com/instana/fleet-helpers/blob/master/rainbow-migration/registerButlerService.sh	Marcel	done
P24	Prepare instanactl database on "cockroachdb-0-eu-west-1"	MB / CS	done
P25	Set all toggles in production "instanactl" database	MB / CS	done
P26	Update instana-saml-cert (use same cert as in SAAS) • create TU - https://saml-instanasaml.instana.io that we can use during Go Live to verify SAML is working	MB / CS / Daniel	done

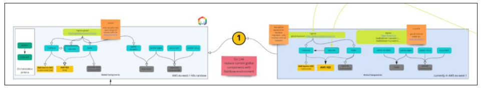
P6 pod listing:

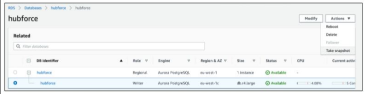
```
kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
accountant-8bf6c6984-w4r4m           1/1     Running   0          115s
bouncer-7c97bf8b4d-998xk             1/1     Running   0          110s
bouncer-7c97bf8b4d-vvbvw             1/1     Running   0          110s
butler-bd56cd47-pkknx                1/1     Running   0          116s
cashier-acceptor-5cc496d949-s67tl    1/1     Running   0          112s
cashier-acceptor-5cc496d949-xtmld    1/1     Running   0          112s
cashier-ingest-5b5598b648-jj6fd      1/1     Running   0          114s
cashier-ingest-5b5598b648-pr56s      1/1     Running   0          114s
cashier-rollup-855888b689-j7kxx      1/1     Running   0          113s
hubforce-global-6995678f8b-v959x     1/1     Running   0          35d
ingress-global-5794ff6fc6-2gchj      1/1     Running   0          111s
ingress-global-5794ff6fc6-qnqxx      1/1     Running   0          111s
```

Last tasks (Monday)

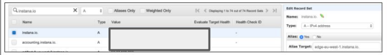
ID	Task	Responsible	Status
L1	Run infrastructure network tests before rollout • https://github.com/instana/fleet-helpers/tree/master/infrastructure-tests	MB / CS	done
L2	Set TTL for **instana.io** to 60sec https://console.aws.amazon.com/route53/home?region=eu-west-1#resource-record-sets:Z20SIWKIE9FAIT	MB / CS	done
L3	Enable rainbow.hcl with production secrets, see https://github.com/instana/instanactl/commit/7e4fc91cc4218d7f92b004a274d77b5bb968aa0a	Marcel	done
L4	Get latest deployed version for: • hubforce - 1.173.44 / master • butler - 1.172.607 / release-172 • bouncer - 1.172.585 / release-172 • cashier-acceptor - 1.172.585 / release-172 • cashier-ingest - 1.172.585 / release-172 • cashier-rollup - 1.172.585 / release-172 • accountant - 1.172.585 / release-172	Marcel	done
L5	Inform Hubforce users about maintenance (Monday afternoon)	Vedran	done
L6	Enable rainbow.hcl with production secrets for hubforce, see https://github.com/instana/hubforce/commit/40deac2ec96c8f0ba1501233f90add79bd8fb672	Vedran	done
L7	Create statuspage maintenance message	CS / MB	done
L8	Stop janitor on orchestration-green	Marcel	done
L9	Create SAAS TU and configure with SAML, **instanasaml-saml**. We will use this TU during the Go Live to verify that SAML is not broken. • Configure SAML	Daniel K.	done

Go Live Steps (Tuesday 6am)



ID	Task	Responsible
G0	Update butler Consul service entry • curl -s http://127.0.0.1:8500/v1/catalog/service/butler | jq . • https://consul-eu-west-1.instana.io/ui/#/eu-west-1/services/butler?filter=butler • https://github.com/instana/fleet-helpers/blob/master/rainbow-migration/registerButlerService.sh	Marcel
G1	Disable TU provisioning in Hubforce	Vedran
G2	Disable TU deploy job in ops-jenkins • https://ops-jenkins.instana.io/job/fleet/job/deploy-tenant-unit/	Marcel
G11	Take snapshot of Aurora DB (hubforce) (about 15 minutes) https://eu-west-1.console.aws.amazon.com/rds/home?region=eu-west-1#database:id=hubforce;is-cluster=false;tab=connectivity 	Schmitzi
G12	Stop Hubforce in SAAS production via Nomad `nomad stop hubforce`	Schmitzi
G13	Deploy Hubforce in K8s rainbow using config from **P4** • https://orchestration-rainbow.instana.io/view/hubforce/	Schmitzi

G14	Stop `butler, accountant, cashier-acceptor, cashier-ingest, cashier-rollup` in SAAS production via Nomad. Update: "We will not stop bouncer."
	`nomad stop butler`
	`nomad stop accountant`
	`nomad stop cashier-acceptor`
	`nomad stop cashier-ingest`
	`nomad stop cashier-rollup`
		MB / CS

G15	Deploy `butler, bouncer, accountant, cashier-acceptor, cashier-ingest, cashier-rollup` in K8s rainbow production via `instanactl` using config from **P1** • https://orchestration-rainbow.instana.io/view/instanactl/job/instanactl-update-global/	
G30	Point instana.io to globalbackend.instana.io in Route53 (**ingress-global** == 2 nginx on K8s) • https://console.aws.amazon.com/route53/home?region=eu-west-1#resource-record-sets:Z20SIWKIE9FAIT • https://console.aws.amazon.com/route53/home?region=us-west-2#resource-record-sets:ZJLKSKP7Z3PK1 	MB / CS
G31	Update reverse proxy (nginx) config to point to K8s rainbow using PR from **P21** • `itermocil global-backend` • `chef-cascade -o 'recipe[fleet-loadbalancer::global-backend]' -W`	MB / CS

G11 screenshot:

RDS > Databases > hubforce > hubforce
hubforce — Modify — Actions ▼ (Reboot, Delete, Failover, Take snapshot)
Related — Filter databases

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity
hubforce	Regional	Aurora PostgreSQL	eu-west-1	1 instance	Available	-	
● hubforce	Writer	Aurora PostgreSQL	eu-west-1c	db.r4.large	Available	4.08%	5 Con

G30 screenshots:

instana.io — A — Aliases Only — Weighted Only — Displaying 1 to 74 out of 74 Record Sets

Name	Type	Value	Evaluate Target Health	Health Check ID
instana.io.	A	[redacted]	-	
accounting.instana.io.	A		-	

Edit Record Set — Name: instana.io. — Type: A – IPv4 address — Alias: ● Yes ○ No — Alias Target: edge-eu-west-1.instana.io.

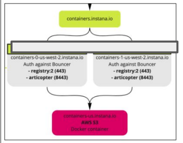
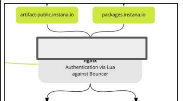
globalbackend.instana.io. A [redacted]

Go/NoGo Testing

ID	Task	Responsible
T1	Hubforce • https://instana.io/portal2/ • Authentication, Logs, ...	Vedran
T2	Login / Authentication (butler) • using stan@instana.io • using own email address • forgot email link	Everybody
T3	Google Single Sign On / SAML • https://saml-instanasaml.instana.io	Daniel K.
T4	2FA	QA
T5	Butler: Tenant Switcher (instana.io/tenantSwitcher)	QA
T6	Check groundskeeper logs for errors • use rollbar and check "SAAS" projects	SRE
T10	Incoming data (should not be impacted since agent keys are loaded from groundskeeper) • acceptors • https://eu-instanaops.instana.io/#/internal/monitoringUnit/sre/acceptors?timeline.to&timeline.fm&timeline.ar=true&timeline.ws=3600000	QA
T11	Incoming data (should not be impacted since agent keys are loaded from groundskeeper) • eum • https://eu-instanaops.instana.io/#/websiteMonitoring/website;websiteId=854_zviATDW1EkZ9-teR2Q/summary?timeline.to&timeline.fm&timeline.ar=true&timeline.ws=3600000	Ben
T12	Incoming data (should not be impacted since agent keys are loaded from groundskeeper) • serverless • QA runs AWS housekeeping in lambda	QA
T13	Butler UMP • usage data (=> check if **accountant** access works) • agent download	QA
	• https://eu-instanaops.instana.io/ump/instanaops/eu/usage/hosts	
T14	Spin up new selfservice TU via website • enable TU provisioning in hubforce • enable jenkins deploy job • sign up for selfservice unit , https://www.instana.com/trial/ • create new internl test unit in hubforce, https://instana.io/portal2/#/dashboard	Vedran, SRE
T15	QA will run auth playbook against newly deployed selfservice TU	QA

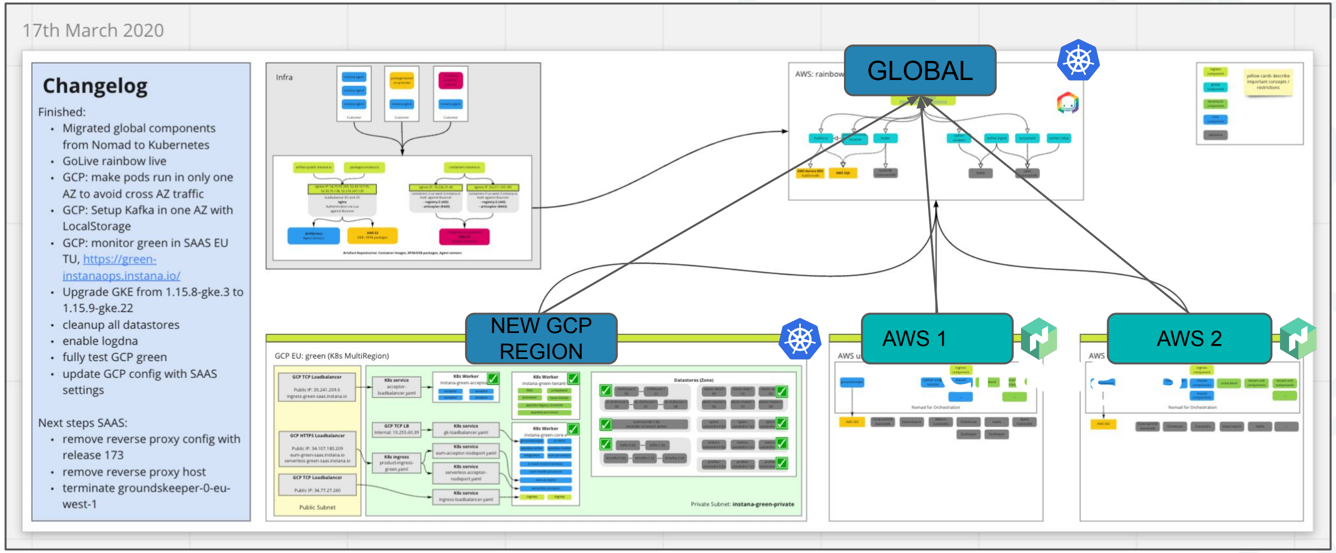
Post GoLive Cleanup

Once K8s Rainbow is up and running, we have several cleanup tasks to get rid of temporary config files and components (i.e. reverse proxy).

ID	Task	Responsible
C1	Artefact Repositories: update `articopter` config use `bouncer` in K8s rainbow • test downloading docker images using onprem agent key 	Marcel
C2	Artefact Repositories: update `loadbalancer` nginx LUA scripts config use `bouncer` in K8s rainbow • test installing package based onprem box using https://ops-jenkins.instana.io/job/delivery/job/deploy-standalone/ • test installing `agent` on new onprem box using single line command 	Schmitzi

Open first GCP Region

- spin up new K8s based GCP region



17th March 2020

Changelog

Finished:

- Migrated global components from Nomad to Kubernetes
- GoLive rainbow live
- GCP: make pods run in only one AZ to avoid cross AZ traffic
- GCP: Setup Kafka in one AZ with LocalStorage
- GCP: monitor green in SAAS EU TU, https://green-instanaops.instana.io/
- Upgrade GKE from 1.15.8-gke.3 to 1.15.9-gke.22
- cleanup all datastores
- enable logdna
- fully test GCP green
- update GCP config with SAAS settings

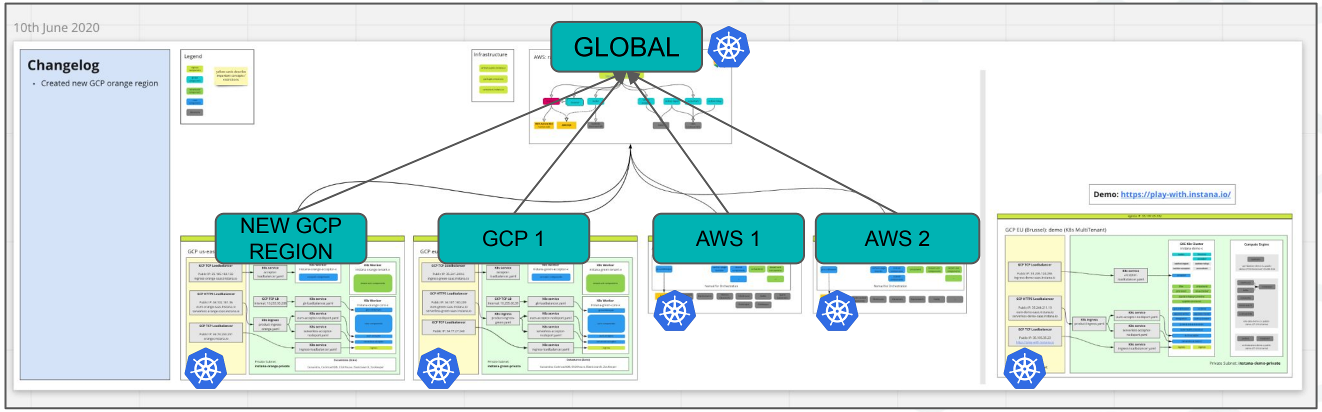
Next steps SAAS:

- remove reverse proxy config with release 173
- remove reverse proxy host
- terminate groundskeeper-0-eu-west-1

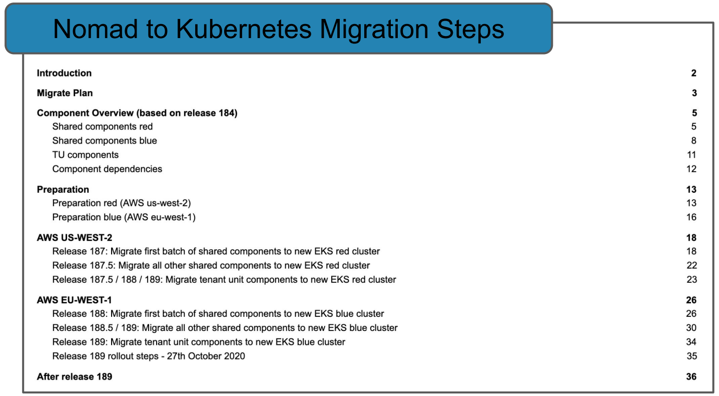
INSTANA
an IBM Company

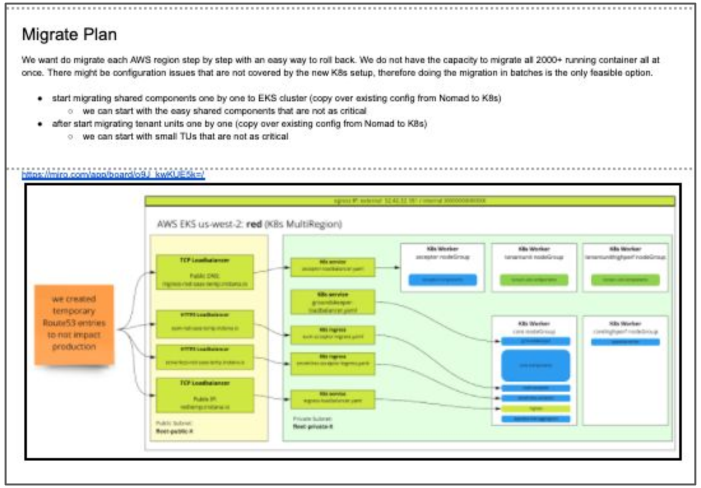
Migrate Nomad to K8s

- Open two more GCP regions
- Migrate Nomad regions to K8s


10th June 2020
Changelog
• Created new GCP orange region
Legend
Infrastructure
GLOBAL
NEW GCP REGION
GCP 1
AWS 1
AWS 2
Demo: https://play-with.instana.io/
GCP EU (Brussel): demo (K8s MultiTenant)

Nomad to Kubernetes Migration Steps



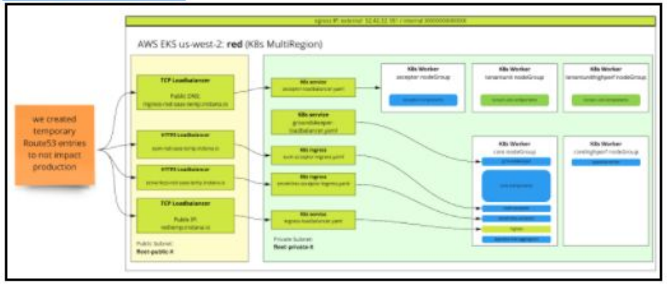


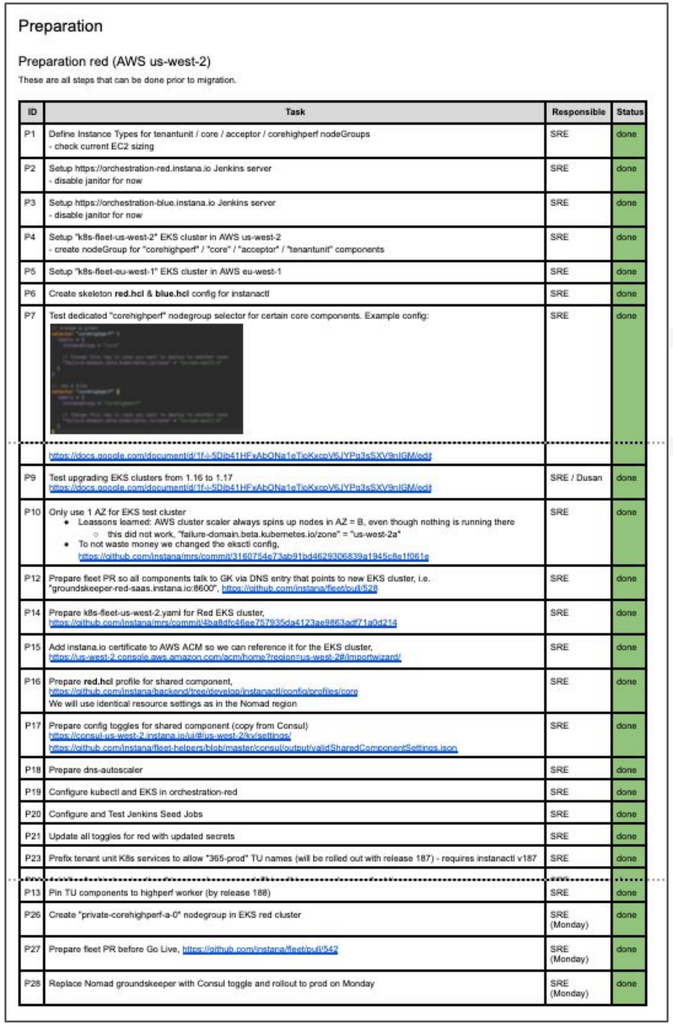
Migrate Plan

We want do migrate each AWS region step by step with an easy way to roll back. We do not have the capacity to migrate all 2000+ running container all at once. There might be configuration issues that are not covered by the new K8s setup, therefore doing the migration in batches is the only feasible option.

- start migrating shared components one by one to EKS cluster (copy over existing config from Nomad to K8s)
  - we can start with the easy shared components that are not as critical
- after start migrating tenant units one by one (copy over existing config from Nomad to K8s)
  - we can start with small TUs that are not as critical

https://miro.com/app/board/o9J_kwKUE5k=/





Preparation

Preparation red (AWS us-west-2)

These are all steps that can be done prior to migration.

ID	Task	Responsible	Status
P1	Define Instance Types for tenantunit / core / acceptor / corehighperf nodeGroups - check current EC2 sizing	SRE	done
P2	Setup https://orchestration-red.instana.io Jenkins server - disable janitor for now	SRE	done
P3	Setup https://orchestration-blue.instana.io Jenkins server - disable janitor for now	SRE	done
P4	Setup "k8s-fleet-us-west-2" EKS cluster in AWS us-west-2 - create nodeGroup for "corehighperf" / "core" / "acceptor" / "tenantunit" components	SRE	done
P5	Setup "k8s-fleet-eu-west-1" EKS cluster in AWS eu-west-1	SRE	done
P6	Create skeleton **red.hcl** & **blue.hcl** config for instanactl	SRE	done
P7	Test dedicated "corehighperf" nodegroup selector for certain core components. Example config: https://docs.google.com/document/d/1f-i-5Dib41HFxAbONa1eTioKxqpV6JYPg3sSXV9nIGM/edit	SRE	done
P9	Test upgrading EKS clusters from 1.16 to 1.17 https://docs.google.com/document/d/1f-i-5Dib41HFxAbONa1eTioKxqpV6JYPg3sSXV9nIGM/edit	SRE / Dusan	done
P10	Only use 1 AZ for EKS test cluster • Leassons learned: AWS cluster scaler always spins up nodes in AZ = B, even though nothing is running there ○ this did not work, "failure-domain.beta.kubernetes.io/zone" = "us-west-2a" • To not waste money we changed the eksctl config, https://github.com/instana/mrs/commit/3160754e73ab91bd4629306839a1945c8e1f061e	SRE	done
P12	Prepare fleet PR so all components talk to GK via DNS entry that points to new EKS cluster, i.e. "groundskeeper-red-saas.instana.io:8600", https://github.com/instana/fleet/pull/528	SRE	done
P14	Prepare k8s-fleet-us-west-2.yaml for Red EKS cluster, https://github.com/instana/mrs/commit/4ba8dfc46ee757935da4123ae9863adf71a0d214	SRE	done
P15	Add instana.io certificate to AWS ACM so we can reference it for the EKS cluster, https://us-west-2.console.aws.amazon.com/acm/home?region=us-west-2#/importwizard/	SRE	done
P16	Prepare **red.hcl** profile for shared component, https://github.com/instana/backend/tree/develop/instanactl/config/profiles/core We will use identical resource settings as in the Nomad region	SRE	done
P17	Prepare config toggles for shared component (copy from Consul) https://consul-us-west-2.instana.io/ui/#/us-west-2/kv/settings/ https://github.com/instana/fleet-helpers/blob/master/consul/output/validSharedComponentSettings.json	SRE	done
P18	Prepare dns-autoscaler	SRE	done
P19	Configure kubectl and EKS in orchestration-red	SRE	done
P20	Configure and Test Jenkins Seed Jobs	SRE	done
P21	Update all toggles for red with updated secrets	SRE	done
P23	Prefix tenant unit K8s services to allow "365-prod" TU names (will be rolled out with release 187) - requires instanactl v187	SRE	done
[illegible]	[illegible]	[illegible]	[illegible]
P13	Pin TU components to highperf worker (by release 188)	SRE	done
P26	Create "private-corehighperf-a-0" nodegroup in EKS red cluster	SRE (Monday)	done
P27	Prepare fleet PR before Go Live, https://github.com/instana/fleet/pull/542	SRE (Monday)	done
P28	Replace Nomad groundskeeper with Consul toggle and rollout to prod on Monday	SRE (Monday)	done

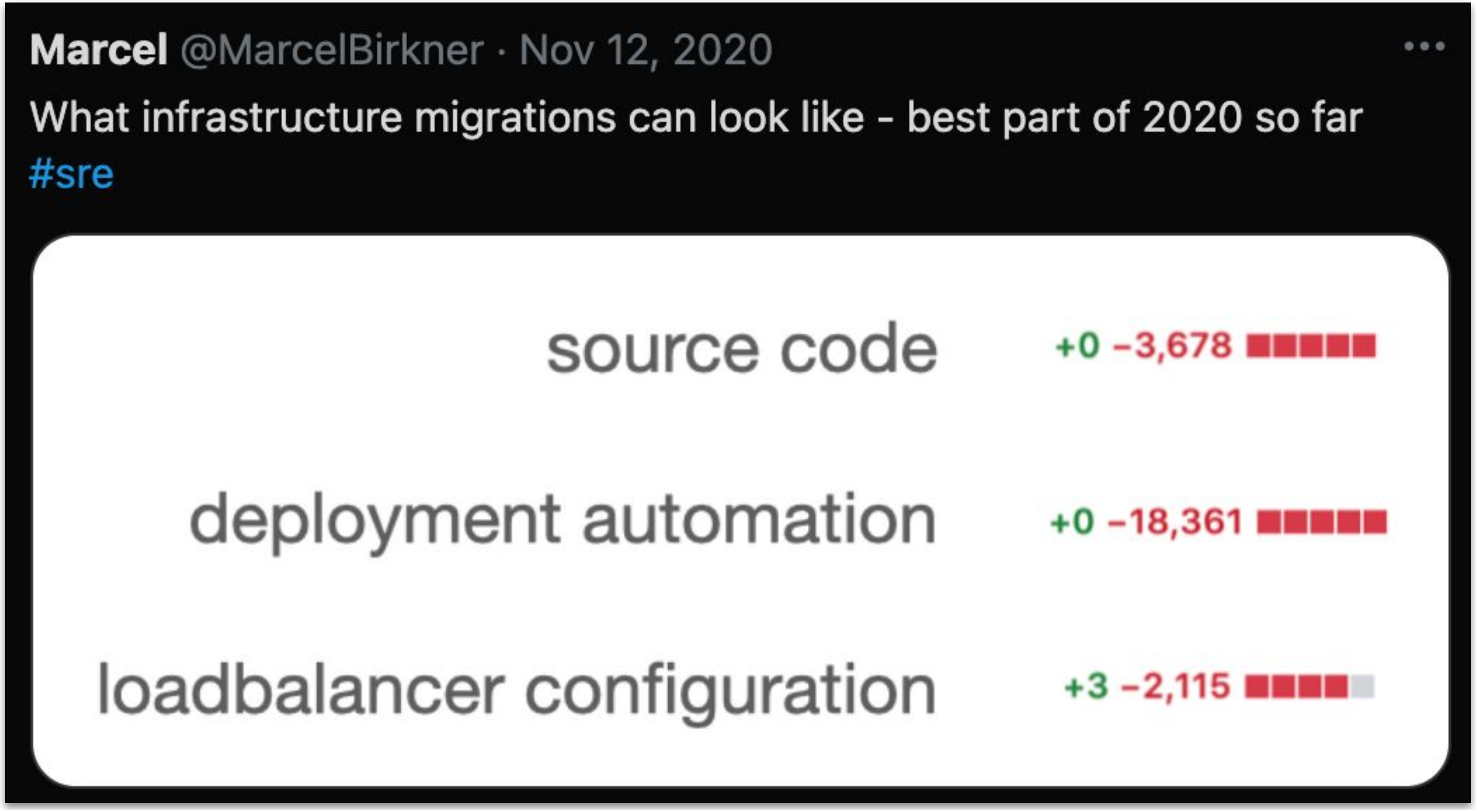
a lot more pages ...

GoLive blueprint

- create plan for infrastructure migration and document all steps (i.e. Miro & Google Docs)
    - **Infrastructure preparation**
        - ID, Task, Responsibility, Status
    - **GoLive steps**
    - **Go/NoGo steps**
    - **Rollback strategy**
- test all steps mentioned above in production-like environment
    - account for DNS timeouts, loadbalancer changes, Elastic IPs (communicate changes early to customers so they can prepare their network egress configuration)
    - stay away from big-bang migrations
    - automate infrastructure tests, so you can verify that new infrastructure works
    - test from various continents (servers in EU and US)
- communicate GoLive plan and gather engineers and QA that help during GoLive
- coordinate rollout with regular releases (bi-weekly @ Instana)
- Do It!

INSTANA
an IBM Company

Project aftermath



Marcel @MarcelBirkner · Nov 12, 2020

What infrastructure migrations can look like - best part of 2020 so far #sre

source code	+0	−3,678	🟥🟥🟥🟥🟥
deployment automation	+0	−18,361	🟥🟥🟥🟥🟥
loadbalancer configuration	+3	−2,115	🟥🟥🟥🟥⬜

Infrastructure improvements

- test coverage for **instanactl**
- flexible deployments across SaaS and On-Premises releases
- networking infrastructure has been simplified
- spinning up new SaaS regions across cloud providers only takes a few days
    - before this was impossible due VPC paring, shared datastores across regions, complex security groups, ...

Unplanned benefits of K8s migration

- Managed K8s in all regions (GCP GKE, AWS EKS, IBM OpenShift)
    - great community and tooling around K8s
    - cluster auto scaler, certificate manager, ...

Meet me in the chat lounge for Q&A

INSTANA

an IBM Company