SKILup℠ DAYS

by: DevOps INSTITUTE

INSTANA

SRE: A day in the life ...

by Marcel Birkner

Bio

Marcel Birkner works as a Staff Reliability Engineer at Instana, an Application Performance Monitoring (APM) solution. He has long experience in software engineering and software automation. Currently he focuses on migrating the existing stack to Kubernetes and reducing overall system complexity.

Abstract

What does a typical day as an SRE look like? In this presentation I will discuss the challenges we face while running a SaaS platform that is used 24 / 7 / 365 around the globe. In doing so, we have embraced the core principles described in the Google SRE handbook. While we are not Google by any means, most of the principles apply to our daily work one way or another. Having a fully distributed team running a distributed system can be quite challenging. In this talk I'll be covering:

- Core SRE principles
- How Instana has applied them to our daily work
- Lessons learned along the way

Who We Are

SRE Team

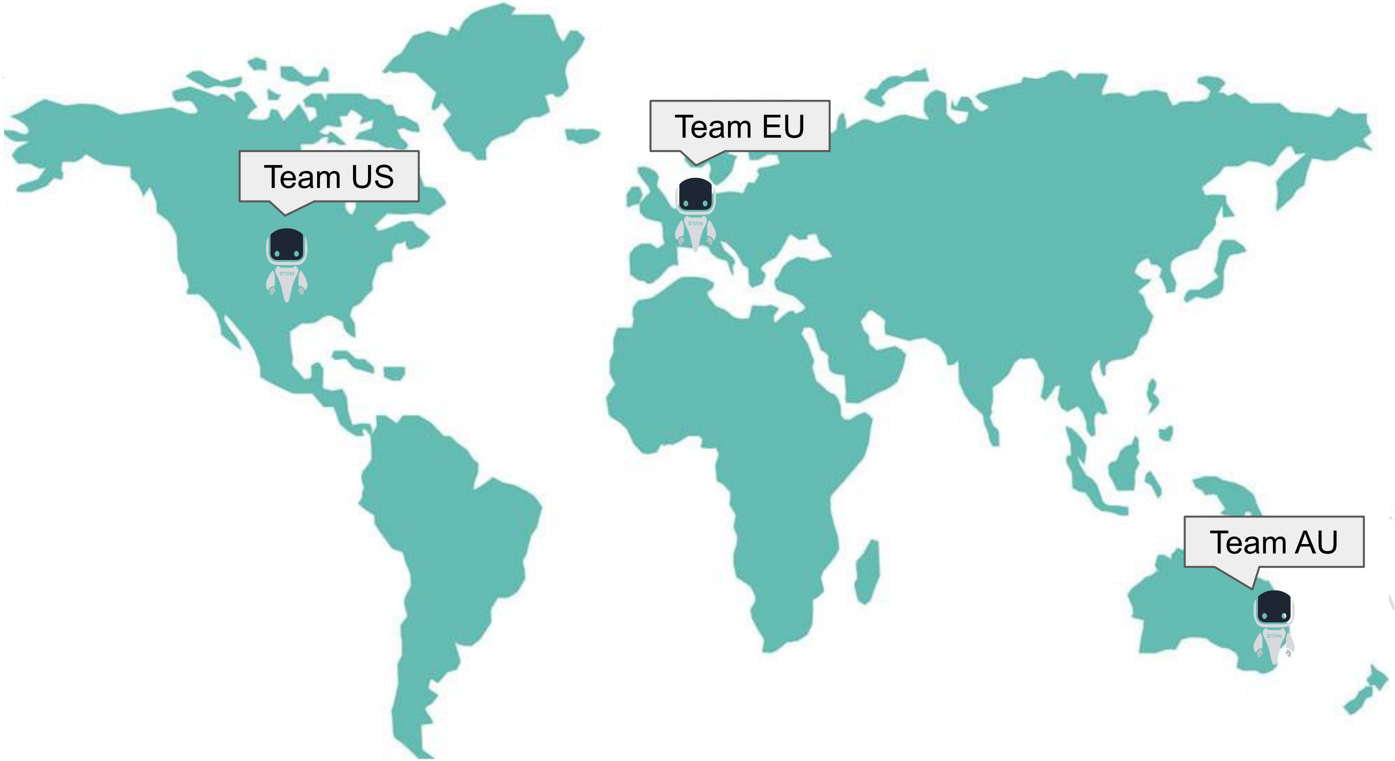
1 Team

3 Time zones

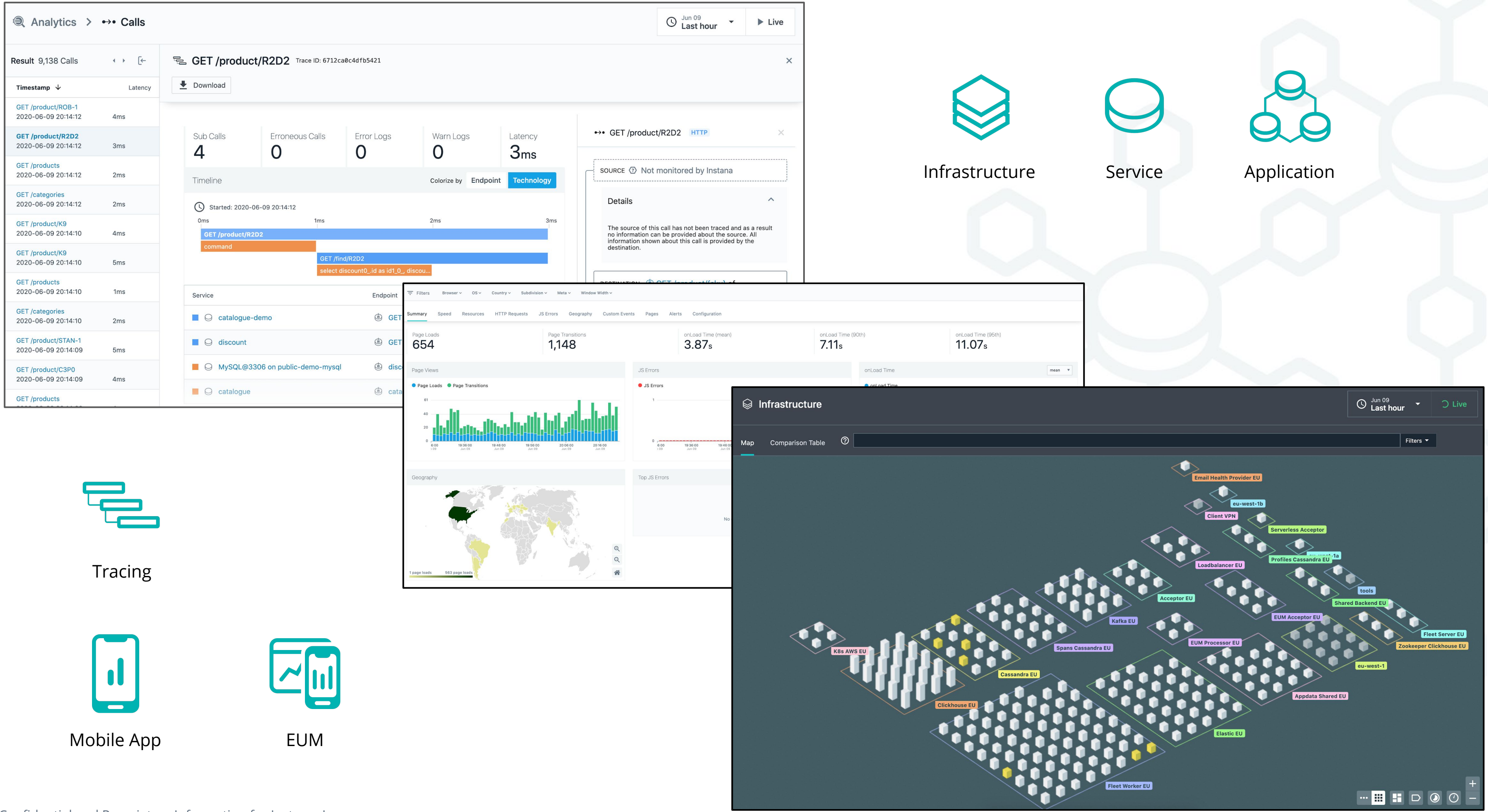
24 / 7 / 365 support

On-call rotation

Members have operations and software engineering background



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

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.

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

Infrastructure

Service

Application

Tracing

Mobile App

EUM

Stats

280 TB / Month
Ingress

8 PB / Month
Cross AZ Traffic

30K+ ECU

8 different datastore
clusters / region

4K+ Containers
Running in SaaS

1M+
TRACES/SEC ANALYZED

300K+
CONTAINERS MONITORED

140K+
JVMS MONITORED

60K+
PODS MONITORED

50K+
HOSTS MONITORED

20K+
FUNCTIONS MONITORED

SaaS Regions

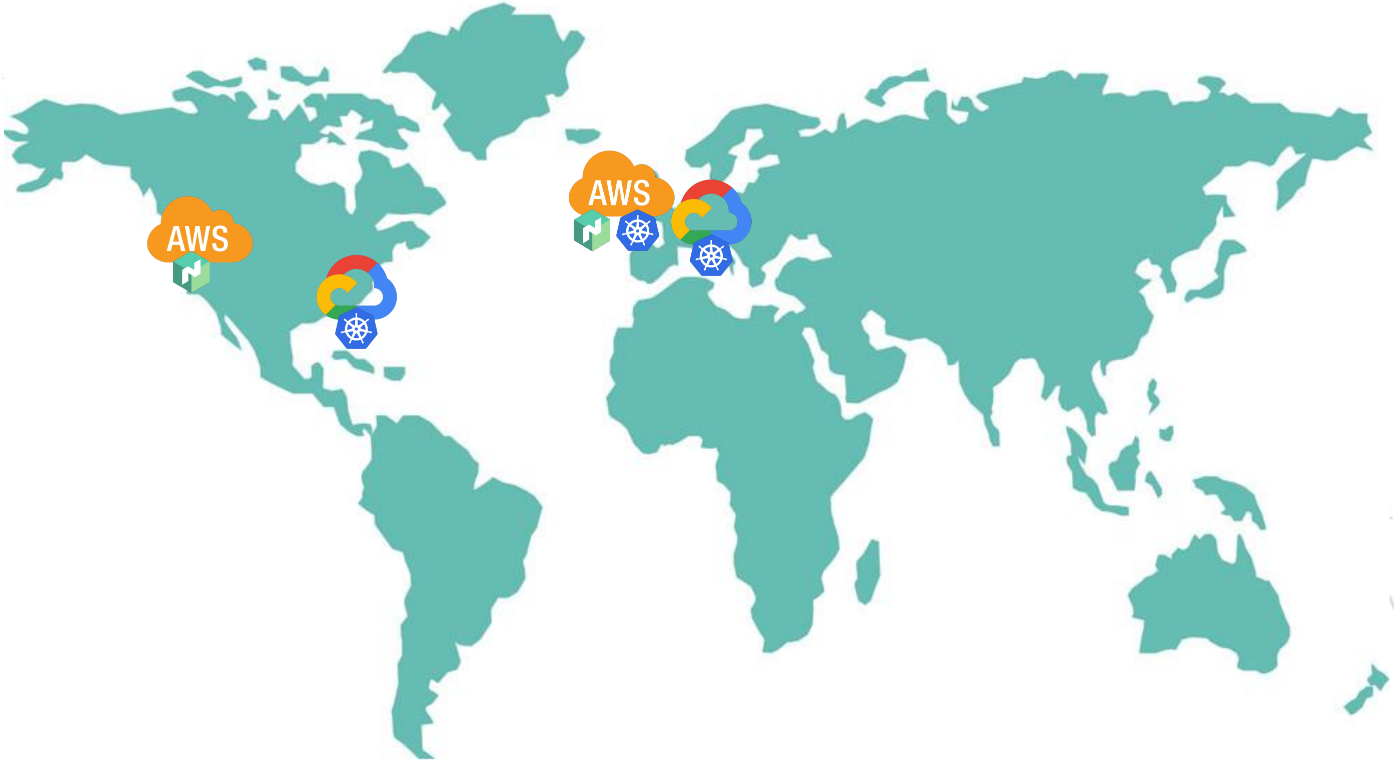
Multi Cloud Strategy

2 x AWS regions

2 x GCP regions

HashiCorp Nomad/Consul

Kubernetes



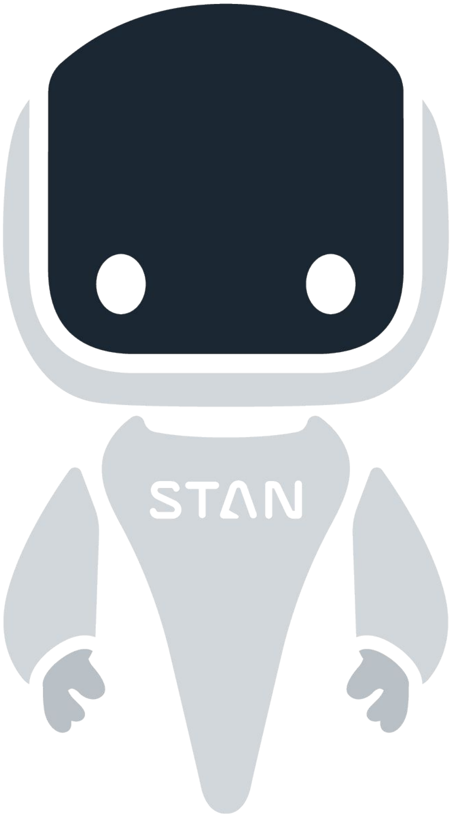
How We Do It

SRE by the book

Automation	Error Budgets	Capacity Planing
Software Development	SLI / SLO / SLA	Maintenance
Cost Planning	RCA	Network Ops
Database Ops	OnCall	Platform Eng / Ops
Dev Support	Post Mortems	...

Planned Day

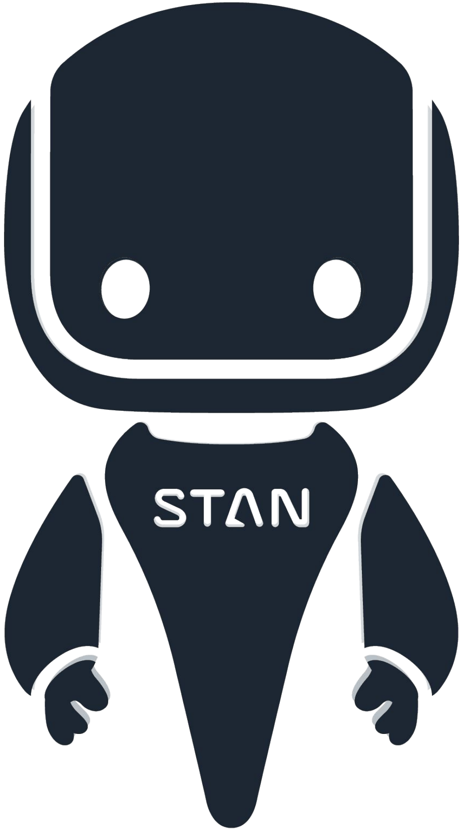
30 min	Handoff Team AU
50%	Tickets/QoS
50%	Project work
30 min	Handoff Team US



Actual Day

- Handoff Team AU
- Alerts
- Ping by Engineering
- Ping by SE / PM
- Ping by CS
- less Project work than planned
- Handoff Team US

Learn to say “No”



Communication is vital

"Something is broken"

Engineering:

"Okay, will have a look"

Sales / CS:

"OMG" => Escalation to CEO => Escalates to VP Eng.

Private Slack Channels
tech-*

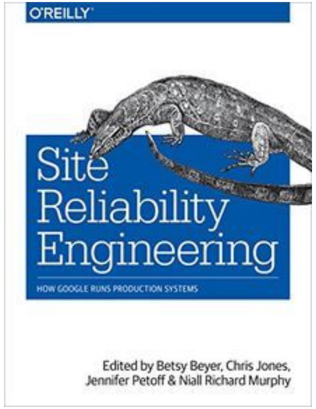
Avoid Panic

SRE Team Priorities

- Quality of Service of SaaS platform
- Platform Security
    - regularly security scans
- Project Work
    - Multi Cloud (AWS & GCP)
    - Cost Management
    - Migrate platform to Kubernetes
    - Upgrade Elasticsearch clusters
    - Integrating new datastore (BeeInstant)
- Support On-Premises
- Developer Support
- Packaging and Delivery

Google SRE Book

Part II: Principles


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Richard Murphy

Embracing Risk

Service Level Objectives

Eliminating Toil

Monitoring Distributed Systems

Release Engineering

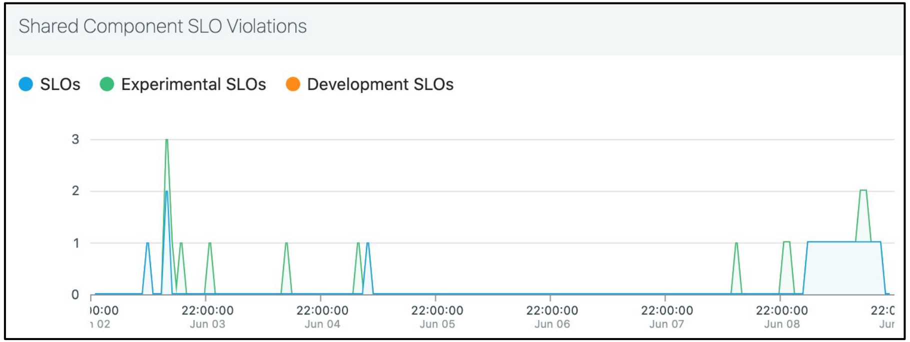
Simplicity

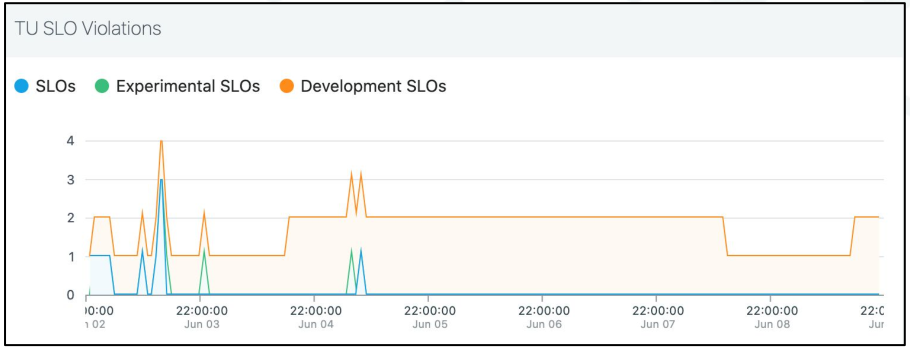
Embracing Risk

- Redundancy / HA / failover
  - datastore clusters across AZ
  - horizontal scaleout of components
- Costs
  - Cost per monitored host
  - K8s / Nomad Orchestration bin-packing
  - Managing TU resources
- Beta Phase for new features
  - Test using internal units
  - Beta customers
- Coming soon: Error Budgets

Service Level Indicators / Objectives

- Custom SLOs for all components in SaaS platform
  - SLO configuration stored and versioned with backend code
  - Updated via REST API after each release
  - Identical across all regions
- Managed by Engineering and SRE





Eliminating Toil

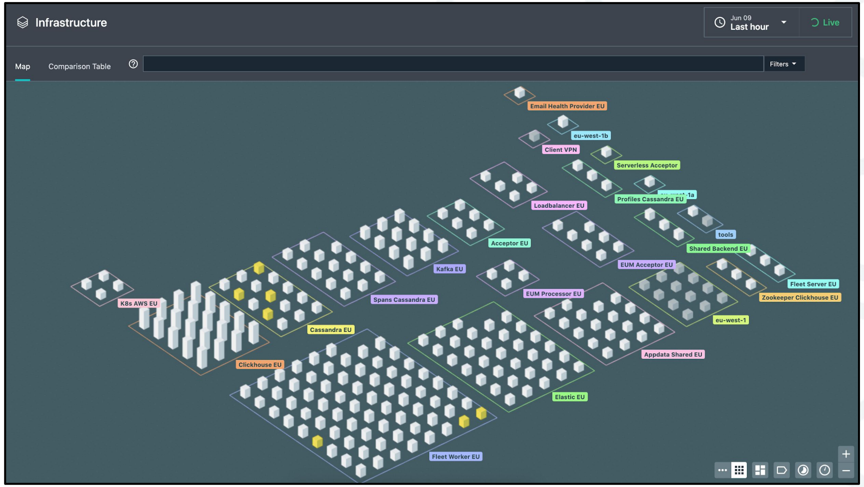
"The moment you have to do something twice, think about automating it"

Spin up new VM	Jenkins + Terraform
Setup / Expand datastore cluster	Chef recipies
Deploy / Update components	Jenkins + instanactl
Run migrations	Jenkins + instanactl
Configure Jenkins Job	Jenkins Job DSL (all jobs are generated)
Configure DNS	instanactl / external-dns (a few DNS entries are manually configured)
Setup GKE cluster	gcloud
Setup EKS cluster	eksctl

Monitoring Distributed Systems

We use Instana to monitor Instana

- Datastores (Cassandra, ClickHouse, CockroachDB, Elasticsearch, Kafka, ZooKeeper, ...)
- Infrastructure Monitoring
- Java DropWizard
- NodeJS
- Automatic Distributed tracing
- Automatic End-User-Monitoring
- Built-in alerting

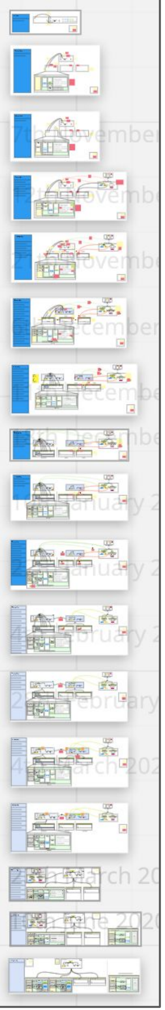


Feedback Loop with PM & Engineering

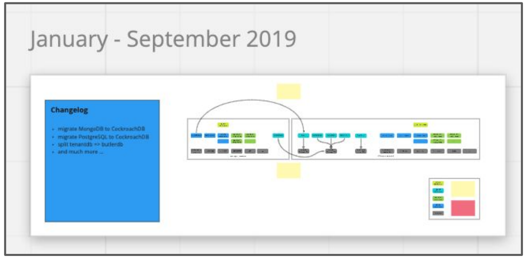
Release Engineering 🚀

- Bi-Weekly Major Releases (Consistency)
- Continuous Release of Beta Features & Improvements & Hotfixes (24 / 7)
- Rotating Release Engineer
  - Knowledge Sharing / Release Engineer Playbook
- Rollut for new K8s environments fully automated
  - instanactl <environment> upgrade
    - check preconditions
    - run migrations
    - upgrade shared and tenant unit containers
    - check postconditions
- Post Mortem after each release / incident
  - improve / automate / refactor processes

Simplicity, Simplicity, Simplicity, …



Automatic Complexity - Infrastructure



January - September 2019

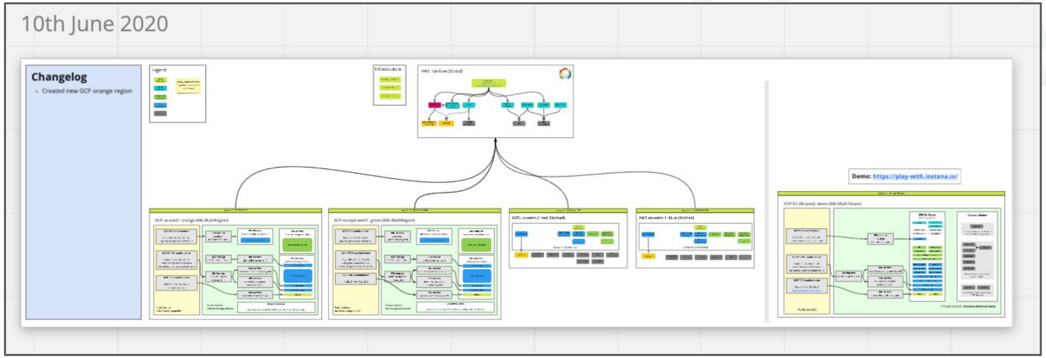
Changelog

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

New Regions

Multi Cloud

Infrastructure automatically becomes more complex over time due to growth and other external factors.



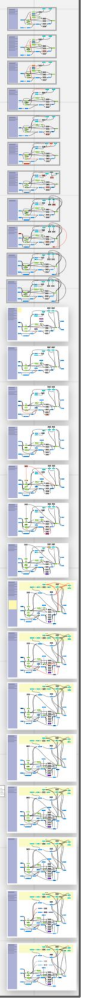
10th June 2020

Changelog

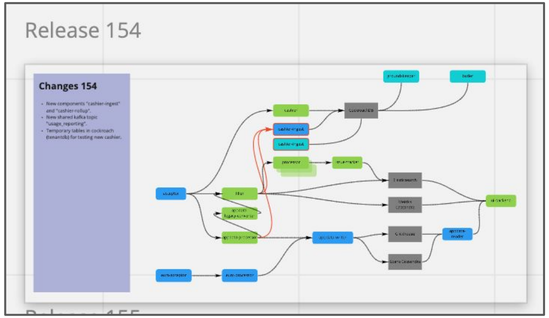
- Created new GCP orange region

Legend
Infrastructure
AWS: rainbow (Global)
GCP us-east1: orange (K8s MultiRegion)
GCP europe-west1: green (K8s MultiRegion)
AWS us-west-2: red (Nomad)
AWS eu-west-1: blue (Nomad)
GCP EU (Brussel): demo (K8s MultiTenant)

Demo: https://play-with.instana.io/



Automatic Complexity - Product



Release 154

Changes 154

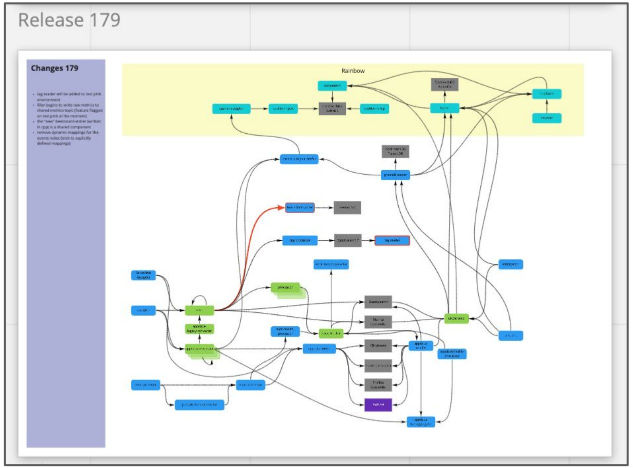
- New components "cashier-ingest" and "cashier-rollup".
- New shared kafka topic "usage_reporting".
- Temporary tables in cockroach (tenantdb) for testing new cashier.

New Features

New Datastores

New Components

System architecture automatically becomes more complex when new features are added over time.



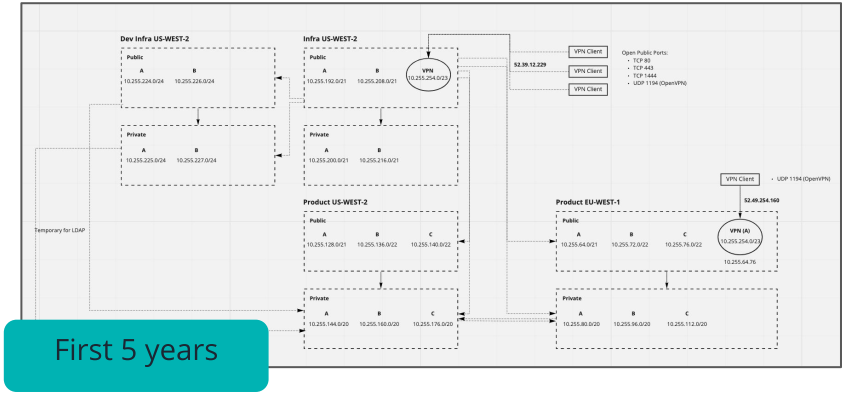
Release 179

Changes 179

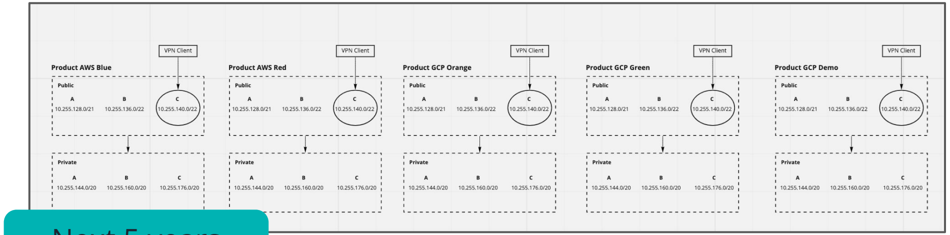
- tag reader will be added to test pink environment
- filer begins to write raw metrics to shared metrics topic (feature flagged on test pink at the moment)
- the "new" beeinstant-writer (written in cpp) is a shared component
- remove dynamic mappings for the events index (stick to explicitly defined mappings)

INSTANA

Work Towards Simplicity



First 5 years



Next 5 years

Plan your infrastructure and network design for growth and simplicity. Keep the overall system as simple possible and only as complex as really needed. This will make your life a lot easier during your typical work day. In times of crisis (i.e. outages) a simple system is easier to understand for all engineers involved to resolve the issue at hand.

Network Design

Infrastructure Design

Common Codebase (SaaS / On-Premises)

up to 2019

Each datastore its migration tool

- 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
- used by SaaS and On-Premises
- single place for migration scripts

Runtimes: Single GoLang Binary

Common Codebase (SaaS / On-Premises)

up to 2019

- separate configuration
- separate packaging (Docker / Packages)
  - SaaS: Docker
  - OnPrem: RPM / DEB
- separate delivery (Ansible / Chef)

Runtimes: Python / Ruby

2020

- same configuration
- same Docker images
- same migration tool
  - instanactl

Runtimes: GoLang Binary & Docker

Supported Operating Systems

Ubuntu 16.04, 18.04   Debian 8.x, 9.x, 10.x   RedHat 7.2+   CentOS 7.x   Amazon Linux 2.x

Lessons Learned

Learn to say “No”

Reduce Complexity

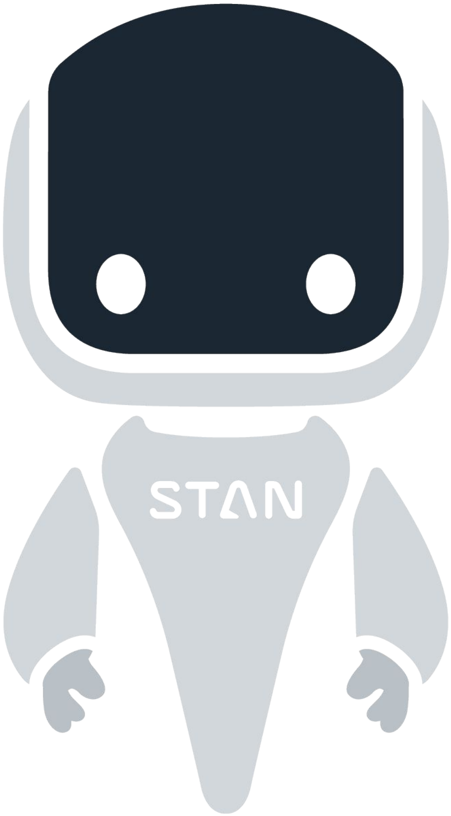
Keep Tooling to a Bare Minimum

Communicate Across Teams

Know Your Tools

Focus and Prioritize Work

Share Knowledge
(SRE runbook, screen recordings, blogs)



Q&A

INSTANA

www.instana.com