Self-healing applications with keptn and Ansible

Jürgen Etzlstorfer
@jetzlstorfer

Ansible Meetup London, June 13th 2019

dynatrace

keptn

What we will talk about

- Why we have to build **keptn**
- What is **keptn** actually?
- How to build **self-healing applications** with **keptn** and **Ansible**

Why we have to build  keptn!

Because Cloud Native Continuous Delivery is a big challenge!

Your App / Container

OPENSHIFT

kubernetes

Pivotal Cloud Foundry

aws

Google Cloud Platform

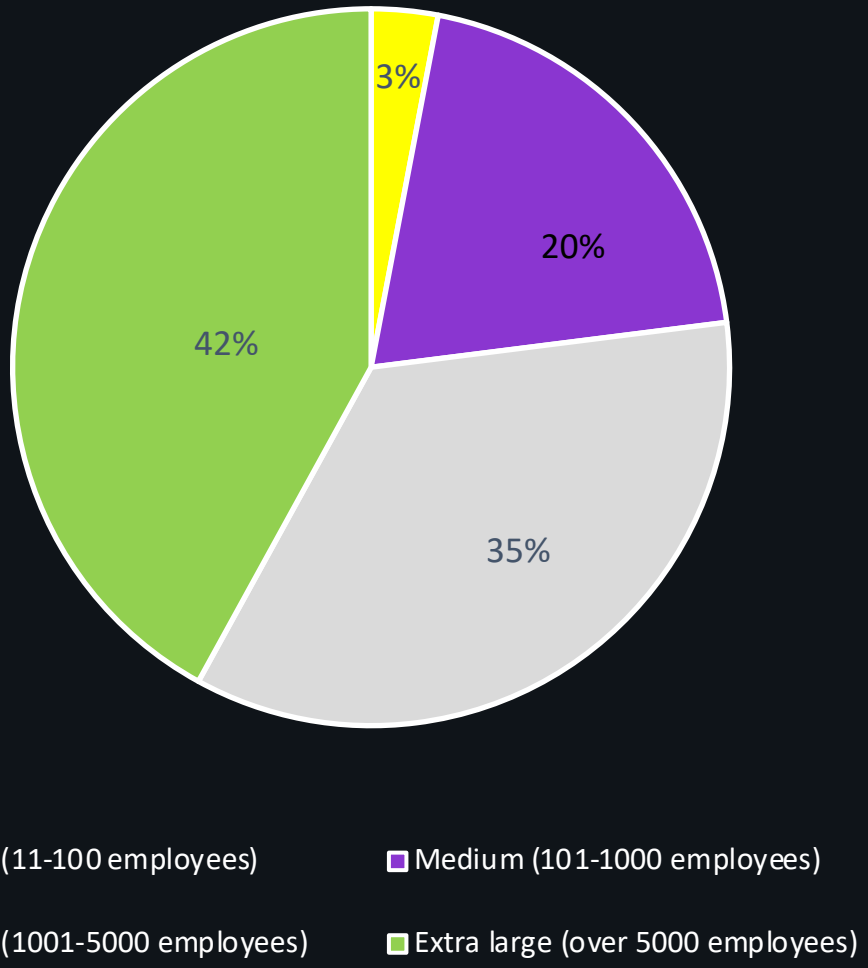
Microsoft Azure

SAP S/4 HANA

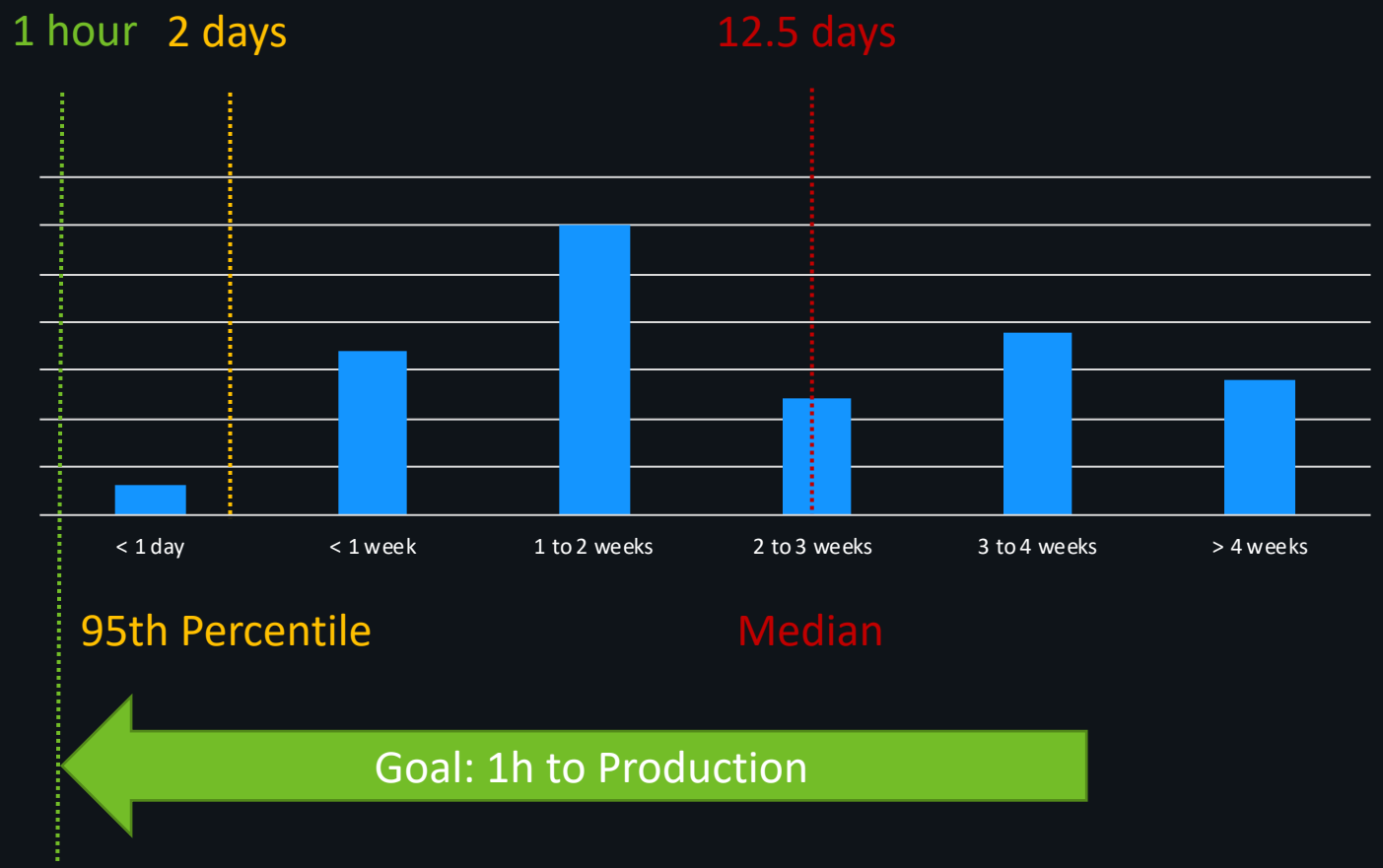
≠

Cloud Native:
Deliver High
Quality Faster

Collecting more evidence: https://dynatrace.ai/acsurvey



Commit Cycle Time: From Dev to Pro



Verdict: The Majority is not delivering high quality faster

	Code to Production (Commit Cycle Time)	Business **Impacting** Deployments	**Per** Production **Deployment**	MTTR (Mean Time to Repair)
95th Percentile	2 days	1 out of 10	0 hotfixes	~4 hours
Median	12.5 days	3 out of 10	3 hotfixes	4.8 days

Evaluate for yourself: https://dynatrace.ai/acsurvey

How do we elevate an organization from Median to 95th percentile!

Faster

97%
deployment lead time

More Frequently

12 to 26
releases per year

Better

75%
production incidents

- Automated Quality (Shift-Left)
  - **Goal**: Stop Bad Code Changes Early & Automated
  - Automated Testing & Quality Gates
- Automated Multi-Stage Deployments (Shift-Right)
  - **Goal**: Increase deployments into **stable** environments
  - Dark, Shadow, Blue/Green Deployments with Auto-Validation
- Automated Operations (Self-Healing)
  - **Goal**: NoOps & Zero-Impact on End Users
  - Automated Remediation & (On-Demand) Scaling

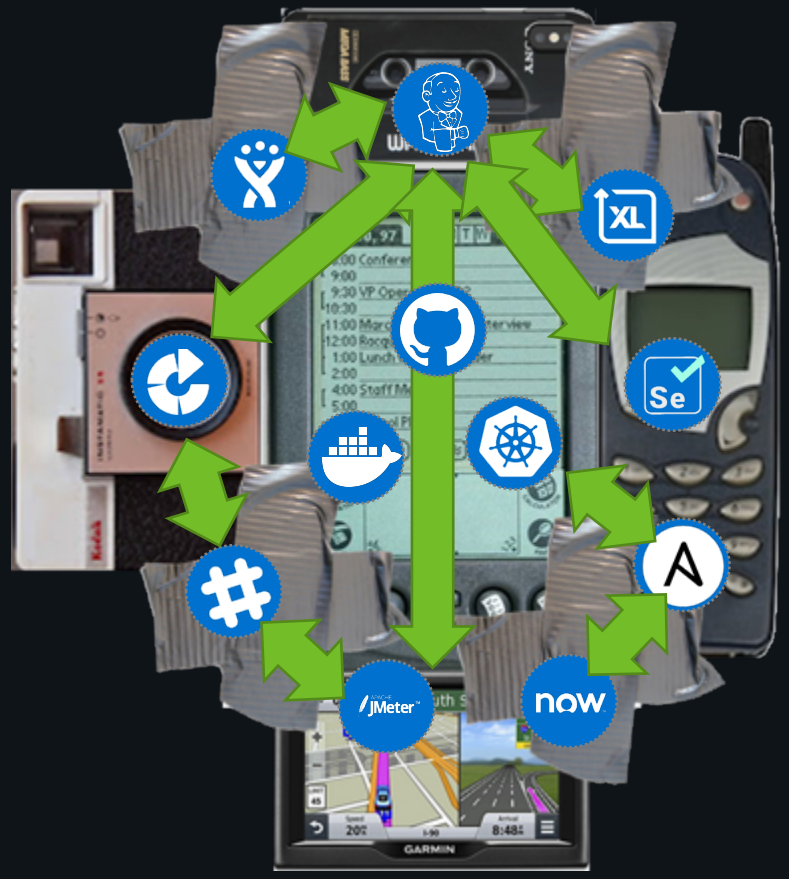
What problems  keptn solves!

Let‘s developers build code vs maintaining pipelines & k8s clusters

Let‘s operations automate run books vs manual firefighting

What we have seen organizations actually do!

- Building Custom Integrations with OpenSource & Commercial tools for
  - Deployment Pipelines
  - Testing Pipelines
  - Auto-Remediation
  - Notifications
  - Auditing



What we have seen organizations struggle with

Quote: „Pipelines seem to become our new future unmanagable legacy code!“

- Teams want to ***stick with existing tools*** to ***protect*** investement
- ***Containing*** lots of ***custom code*** for tool integration, error handling, logging, ...
- ***Getting harder*** to ***maintain*** the more tools get integrated
- ***Pipelines*** becoming more ***complex*** requiring ***dedicated teams***
- ***Uncoordinated*** deployments between pipelines resulting in ***unstable*** Environments

What  keptn is and how it works!

Enterprise-grade framework for shipping and running cloud native apps

keptn from 10000ft

Core capabilities


Automated multistage unbreakable delivery pipelines


Self-healing blue/green deployments


Event-driven runbook automation

Design Principles


GitOps-based collaboration


Operator patterns for all logic components

Monitoring and operations as code


Built on and for Kubernetes


Event-driven and serverless

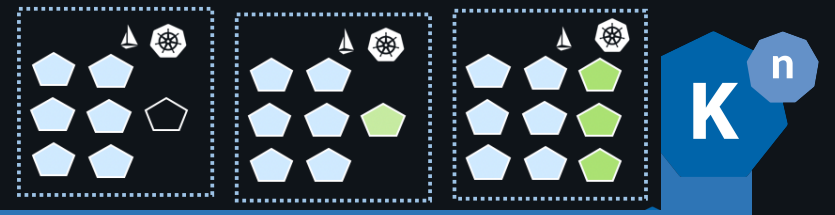

Pluggable tooling

keptn – conceptual architecture

Core

keptn Autonomous Cloud Control Plane

Environment Definition (shipyard file)



Services

GitOps

Container Registry

Continuous Delivery

AIOps

Operations Automation

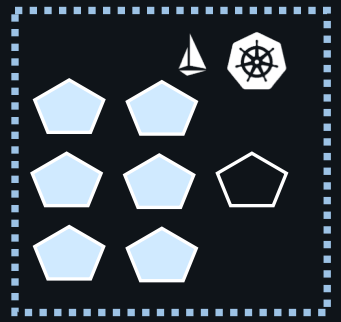
Test Automation

ChatOps

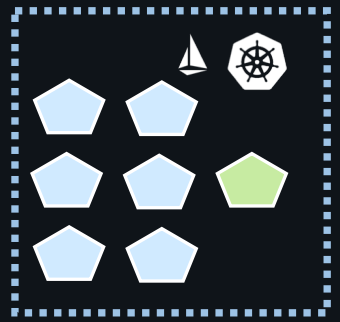
Data Provider

Platform

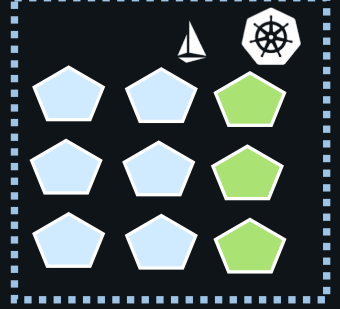
Dev Namespace



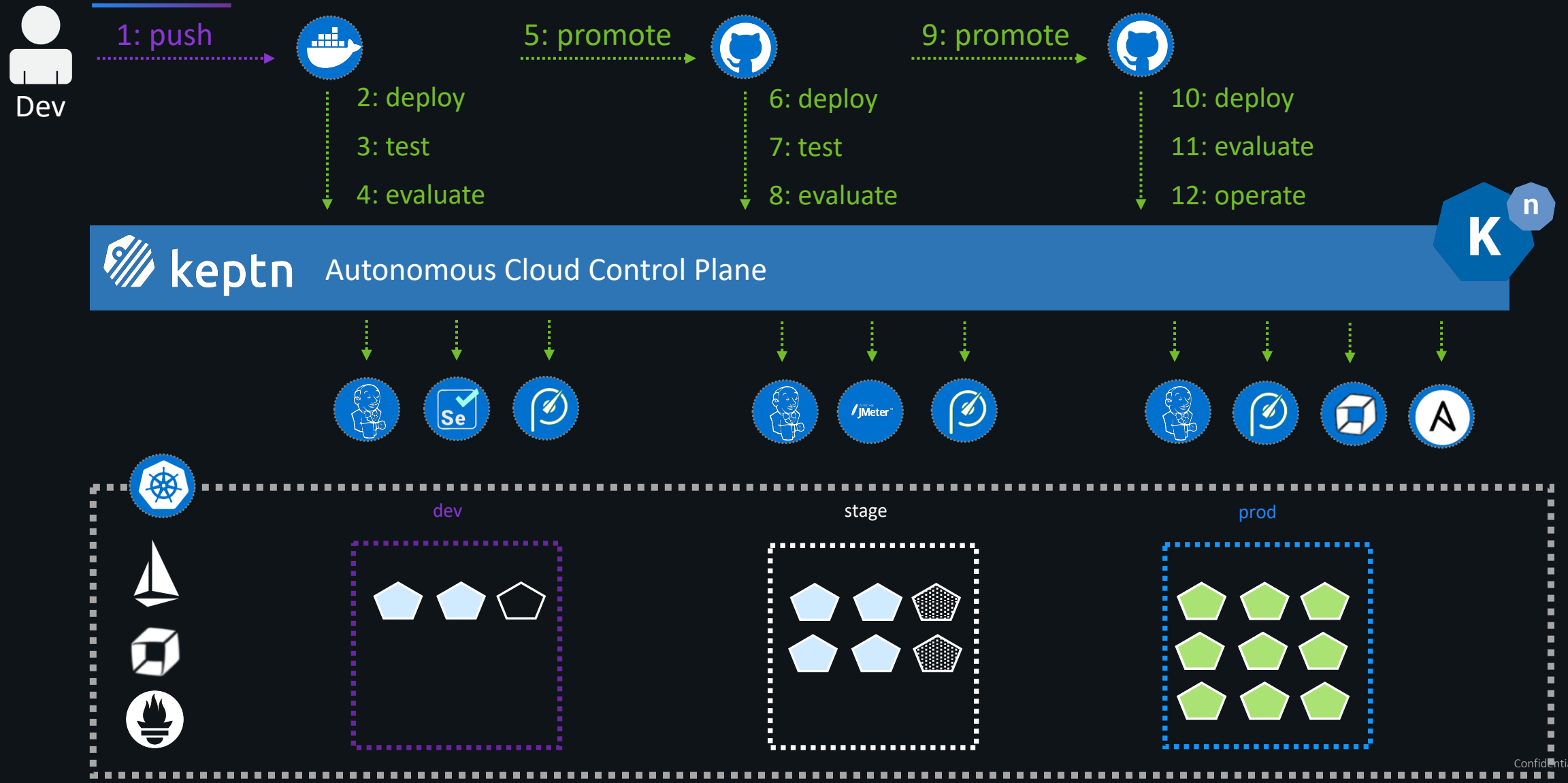
Staging Namespace



Production Namespace



keptn – Shipping through Unbreakable Continuous Delivery Pipelines

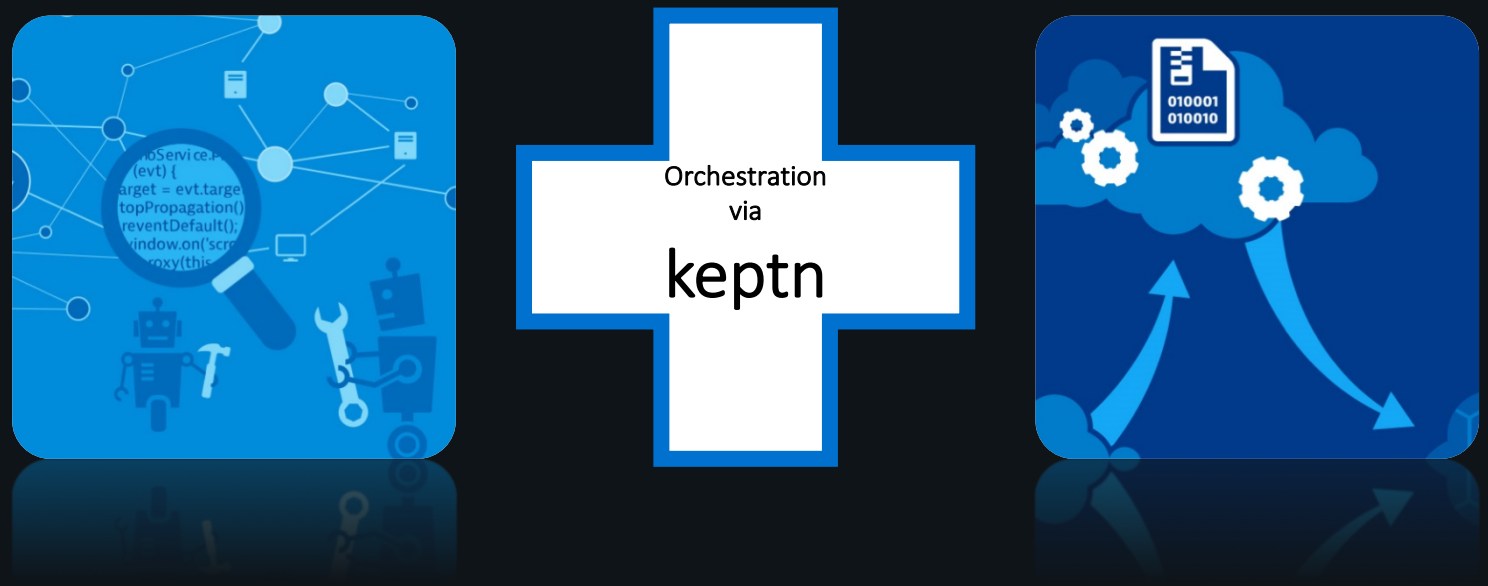


How to build self-healing applications with  keptn and Ansible

How to build your auto-remediation workflow

Auto-remediation building blocks

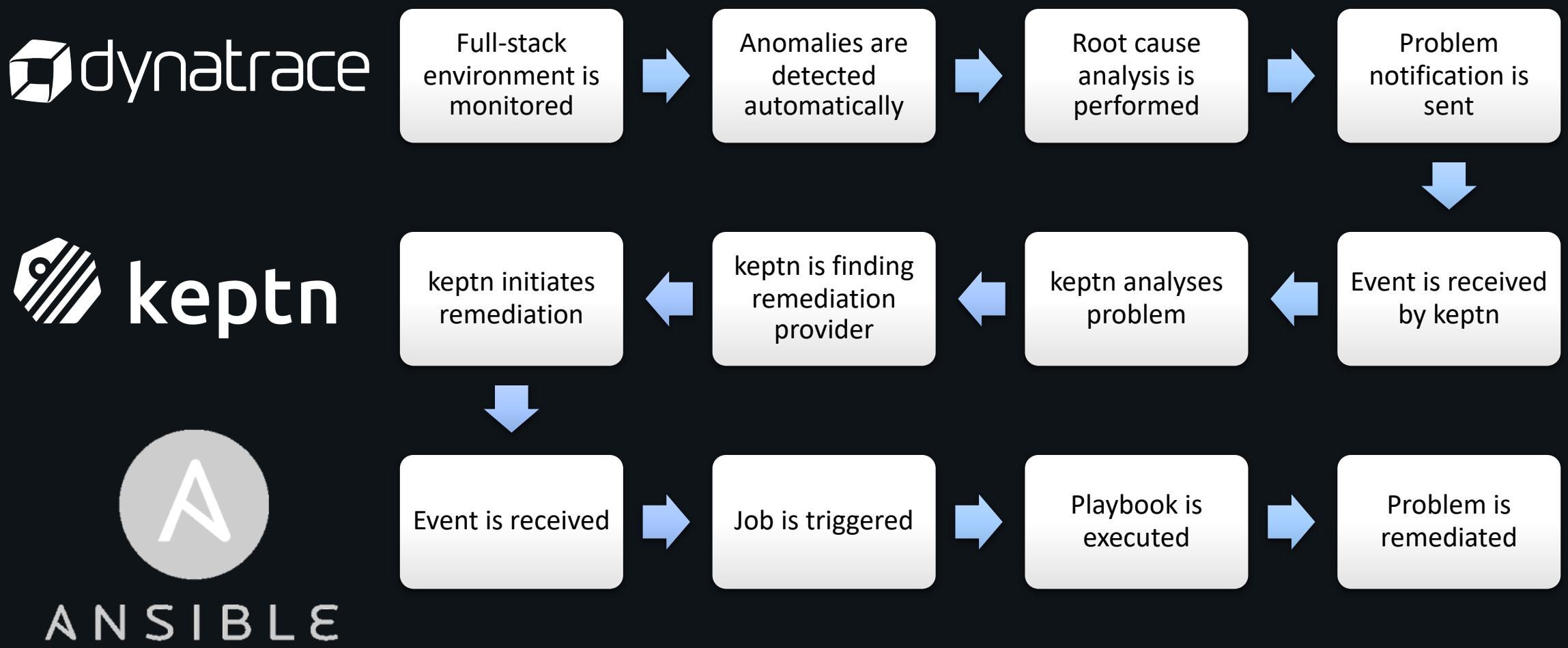
- **Monitoring**: know what's going on in your applications
  - End-to-end
  - Full-stack – fully integrated in production
- **Automation/Execution**: perform mitigation/remediation actions
  - Access to all systems



Automation with Ansible (Tower)

- APIs are key to enable automation
- Ansible Tower makes extensive use APIs internally and exposes them also externally
- Ansible playbooks are scripts that are executed from a central host on different machines
  - Multiple OS are supported
  - Idempotent
- Playbooks can be orchestrated in workflows and job templates

Workflow


dynatrace
Full-stack environment is monitored
Anomalies are detected automatically
Root cause analysis is performed
Problem notification is sent
keptn
Event is received by keptn
keptn analyses problem
keptn is finding remediation provider
keptn initiates remediation
ANSIBLE
Event is received
Job is triggered
Playbook is executed
Problem is remediated

„Demo“ Time


keptn

Want to learn more about keptn?

- Learn how to use keptn: https://keptn.sh/docs/
- Clone, fork, star, contribute to keptn: https://github.com/keptn/
- Develop your own keptn-services: https://github.com/keptn/keptn-service-template
- Join our community (slack invite, meetings): https://github.com/keptn/community
- Follow us @keptnproject
- Fill out our survey: https://dynatrace.ai/acsurvey

- Reach out with any requests:
  - juergen.etzlstorfer@dynatrace.com
  - @jetzlstorfer

More resources

- Joint Dynatrace+Ansible webinar https://github.com/dynatrace-innovationlab/ansible-webinar/
- Set up Ansible Tower with Dynatrace to enable your self-healing applications https://www.dynatrace.com/news/blog/set-up-ansible-tower-with-dynatrace-to-enable-your-self-healing-applications/
- Self-healing: Ansible Tower fixes Dynatrace-detected problems in real time https://www.dynatrace.com/news/blog/self-healing-ansible-tower-fixes-dynatrace-detected-problems-in-real-time/
- Enable self-healing applications with Ansible and Dynatrace https://www.ansible.com/blog/enable-self-healing-applications-with-ansible-and-dynatrace (Ansible) or https://www.dynatrace.com/news/blog/enable-self-healing-applications-with-ansible-and-dynatrace/ (Dynatrace)

dynatrace.com