On-Call Like a King

How do we utilize Chaos Engineering to become better cloud native engineers?

Eran Levy
@levyeran
https://levyeran.medium.com

Introduction

Eran Levy

Director of Software Engineering @Cyren

https://levyeran.medium.com

@levyeran

WIFM?

We leveraged Chaos Engineering principles to achieve other things besides it’s main objectives -

You will learn what we have done to train our engineers on cloud native practices, tooling

and bring confidence while responding to production failures

So what is this buzzword - “Cloud Native” ?

“Cloud native technologies empower organizations to build and run **scalable** applications in modern, **dynamic** environments such as public, private, and hybrid clouds.

Containers, service meshes, microservices, immutable infrastructure, and declarative APIs **exemplify** this approach.

These techniques enable **loosely coupled** systems that are **resilient, manageable, and observable**. Combined with robust automation, they allow **engineers to make high-impact changes** frequently and predictably with minimal toil.”

CNCF Cloud Native Definition v1.0 (https://github.com/cncf/toc/blob/master/DEFINITION.md)

You might be interested in reading my post to read further my view around this topic -

“The Cloud Native Engineer: The engineer evolution at a glance”

Eran Levy
Jun 1, 2020 · 8 min read · Listen

credit: cloudelicious.net

The Cloud Native Engineer: The engineer evolution at a glance

@levyeran

We embrace these changes...

- Take an **end-to-end ownership** of deliveries and enhance velocity.
- Closer to the product and the customer needs **- business impact**!

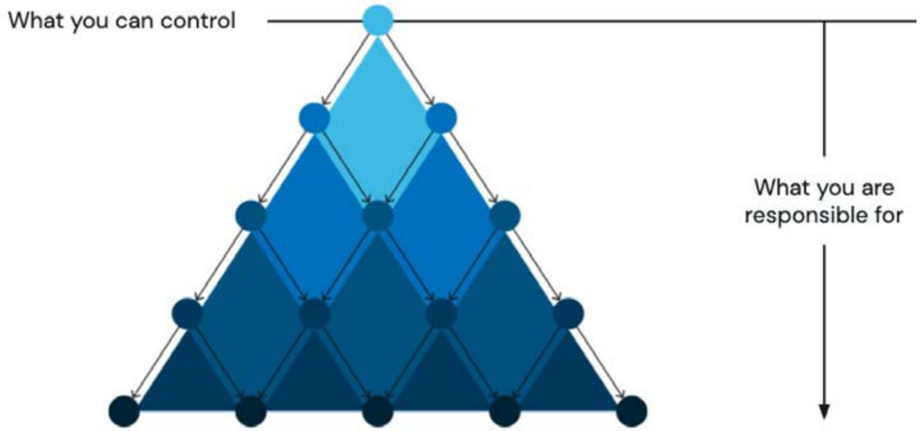
Transition in engineering mindset –

we ship products and not just code!

“Deep systems”

As engineers we usually ship

part of a larger piece of software



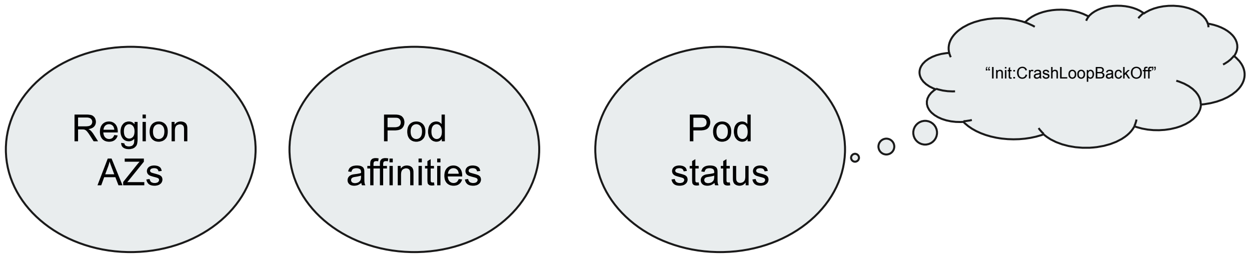
Source: https://lightstep.com/deep-systems/

As engineers we face more challenges

That we didn’t have to deal with before…

You are on-call, some back pressure is burning your SLO targets -
33% of your deployment could not reschedule due to lack of node availability in your cluster

hmmmm…



@levyeran

Being cloud native engineer is fun!

But also challenging...

YAML

HELM

Sidecar

Client-side Load Balancing

Ingress

@levyeran

Talking about Cloud Native without mentioning...

Fallacies of distributed computing:

1. The network is reliable;
2. Latency is zero;
3. Bandwidth is infinite;
4. The network is secure;
5. Topology doesn't change;
6. There is one administrator;
7. Transport cost is zero;
8. The network is homogeneous.
9. We all trust each other.

https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing#The_fallacies

What did we do to cope with these challenges?

We utilized Chaos Engineering for that purpose!

Chaos Engineering

“Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system’s capability to withstand turbulent conditions in production” (https://principlesofchaos.org)

We leverage Chaos Engineering principles to achieves other things besides it’s main objective

“On-Call Like a King”

Main objectives:

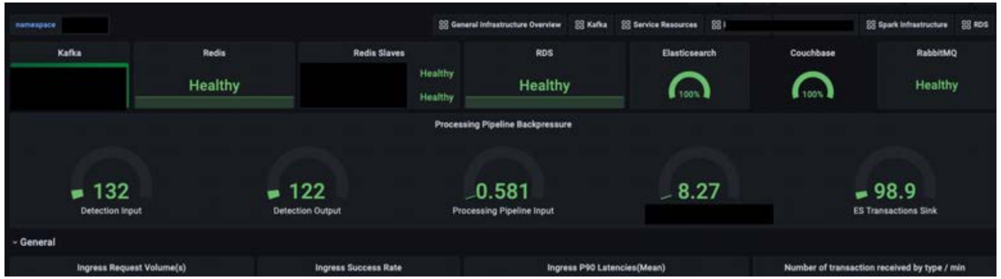
(1) Production failures exercising;
(2) Cloud native practices, tooling and advance knowledge

Source: workshop slides

Before we drill down to the workshop details

Let me share with you how do we do on-call?

The on-call toolbox



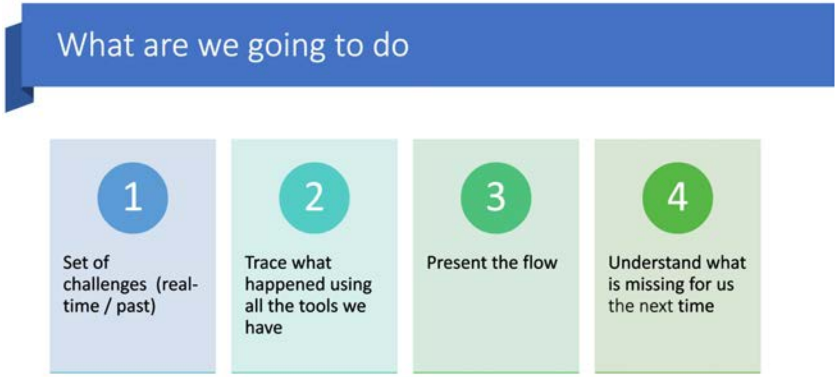
JAEGER

Kibana

@levyeran

How are the workshop sessions composed?

1 - Intro & Motivation


What are we going to do

1 Set of challenges (real-time / past)

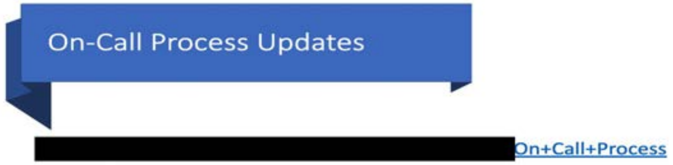
2 Trace what happened using all the tools we have

3 Present the flow

4 Understand what is missing for us the next time

Source: workshop slides

2 - Great opportunity to share important stuff


On-Call Process Updates

[redacted]On+Call+Process

Source: workshop slides

XSOAR Troubleshooting

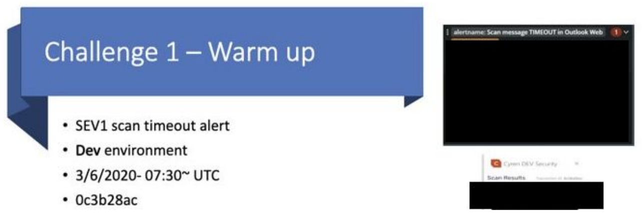
Issues Created Last 24 hours [redacted]

[redacted]/XSOAR

Source: workshop slides

@levyeran

3 - Work on 2 (max) incidents simulation - 60 mins.



Challenge 1 – Warm up

- SEV1 scan timeout alert
- Dev environment
- 3/6/2020- 07:30~ UTC
- 0c3b28ac

alertname: Scan message TIMEOUT in Outlook Web

[redacted]

Cyren DEV Security

Scan Results

[redacted]

Source: workshop slides

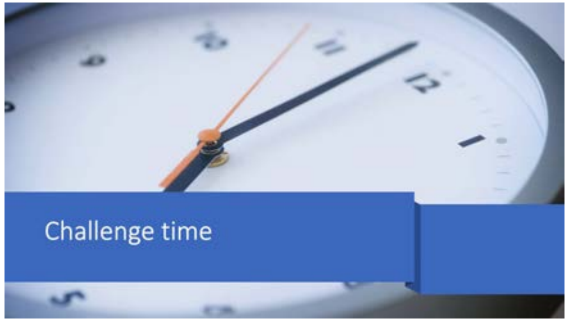
Close to real life production scenarios as possible

Prepare an experiment in advance

We started using LitmusChaos just lately and there are many others such as: Gremlin, ChaosNative and more…

If you don't have one in place, **start manually** - this is how we started 2 years ago

Challenge



SEV1 Service Kafka Lag

[redacted] -IM 9:23 AM
bserving SEV1

Lag by Consumer Group

[redacted]

[redacted]

Source: workshop slides

@levyeran

Give them time

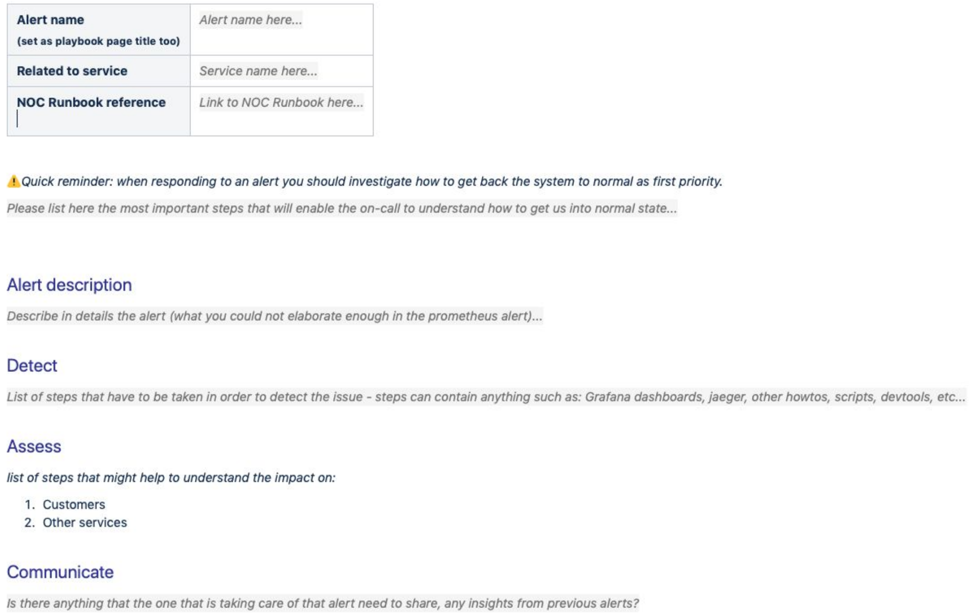
Ask them to think on the customer impact

Pause & encourage to ask questions

- Ask somebody to show the progress
- Ask somebody to show the tools in use
- Drive knowledge sharing

Adapt your incident playbooks

Playbook Template



Alert name **(set as playbook page title too)**	*Alert name here...*
Related to service	*Service name here...*
NOC Runbook reference	*Link to NOC Runbook here...*

⚠️*Quick reminder: when responding to an alert you should investigate how to get back the system to normal as first priority.*

Please list here the most important steps that will enable the on-call to understand how to get us into normal state...

Alert description

Describe in details the alert (what you could not elaborate enough in the prometheus alert)...

Detect

List of steps that have to be taken in order to detect the issue - steps can contain anything such as: Grafana dashboards, jaeger, other howtos, scripts, devtools, etc...

Assess

list of steps that might help to understand the impact on:

1. Customers
2. Other services

Communicate

Is there anything that the one that is taking care of that alert need to share, any insights from previous alerts?

Drive conversations by asking questions

Be a moderator - time is running fast

Point your finger on different aspects

Resolution - ask somebody to present end-to-end

Record the meeting & Share notes

Chaos Engineering for training is a pretty nice tool

@levyeran

Summary

- Great playground - make sure you can experiment in a real environment
- Start quick & dirty - if you don't have the tools, start manual simulation
- Measure how these sessions help people & what can be adapted?
- It can be a great training source for engineers that just onboarded

@levyeran

Thank you!

Eran Levy
@levyeran