



mainframe

We are a 3 year-old pan-European MMO studio



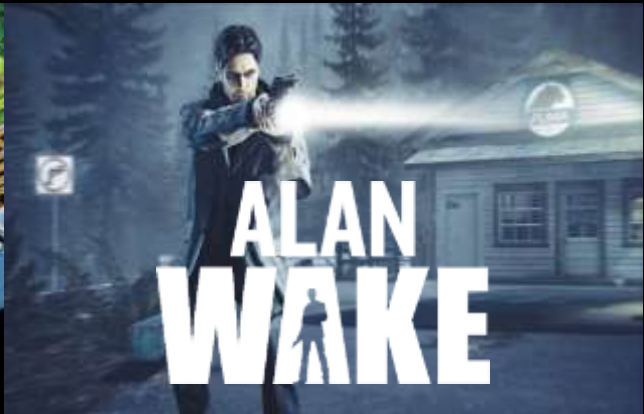
A team of MMO, AAA & Mobile veterans

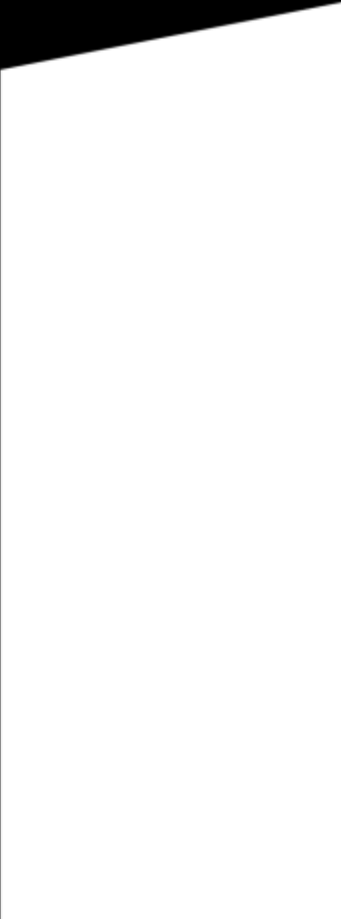


We are a fully distributed pan-European studio with offices in **Helsinki**, **Reykjavík**, and **Paris**.

We support both remote and in-person work, following one simple rule: **work where you are most productive!**

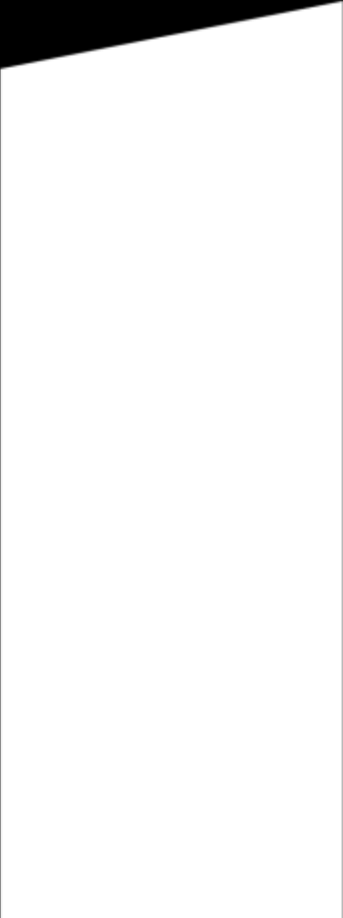
With two decades of shipped games on all platforms



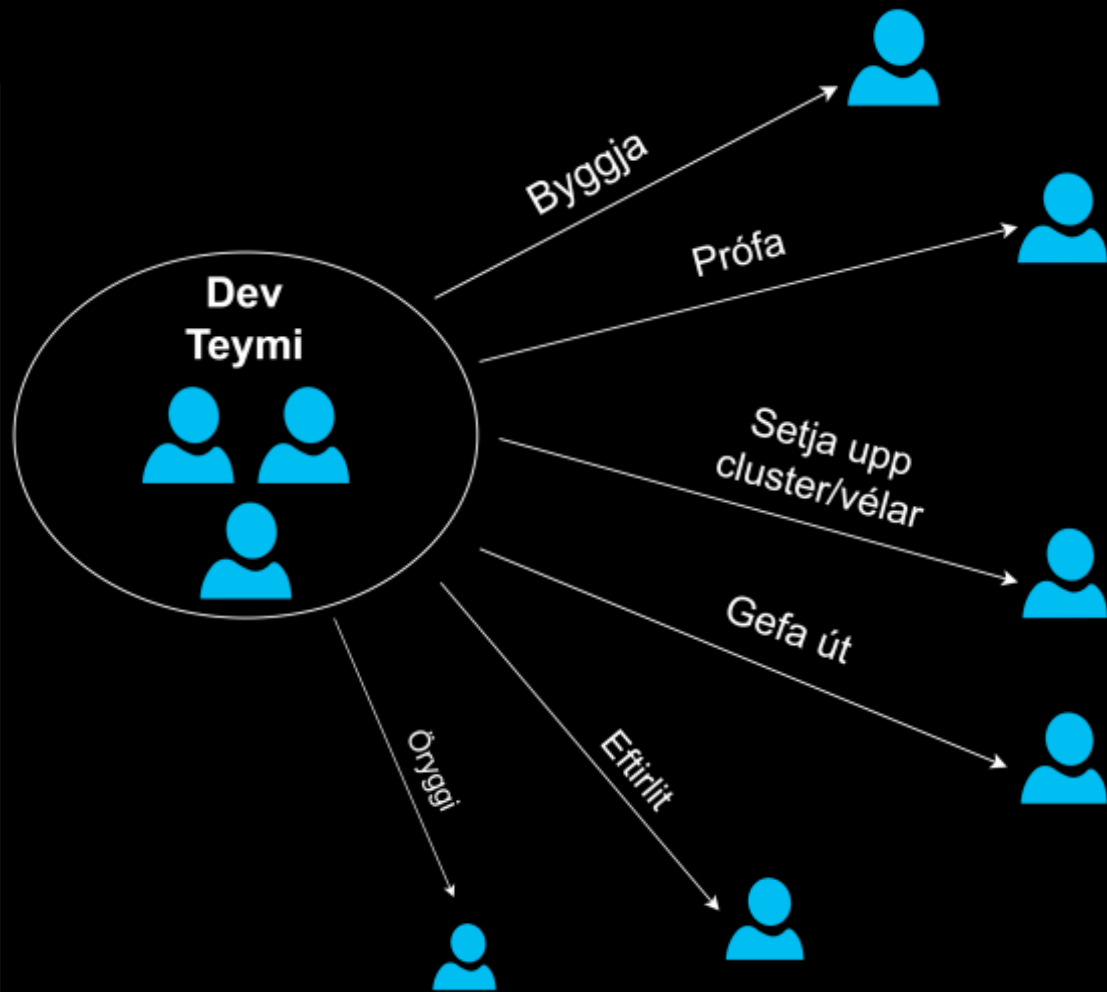


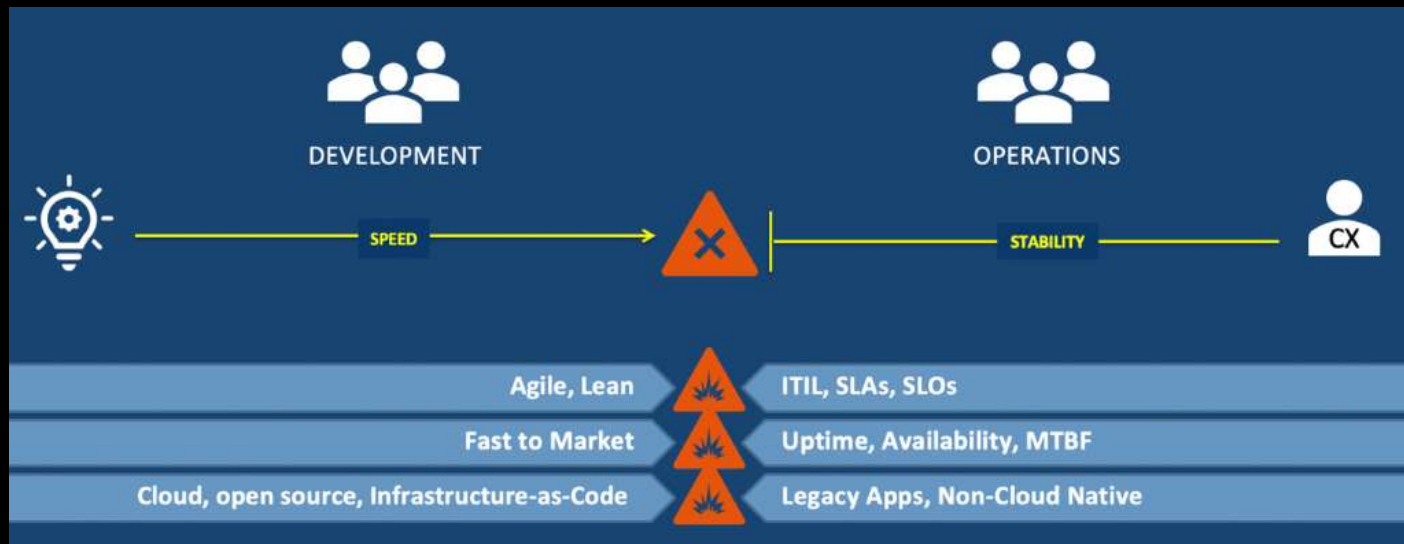
Our Mission:

**Set on building a virtual world where
millions will live, for decades.**



**Hvaðan komum
við? Af hverju
þurfti eitthvað
DevOps drasl?**





I DON'T KNOW WHO YOU ARE

**BUT IF YOU DEPLOY ON A FRIDAY, I
WILL FIND YOU AND I WILL KILL YOU**

imgflip.com

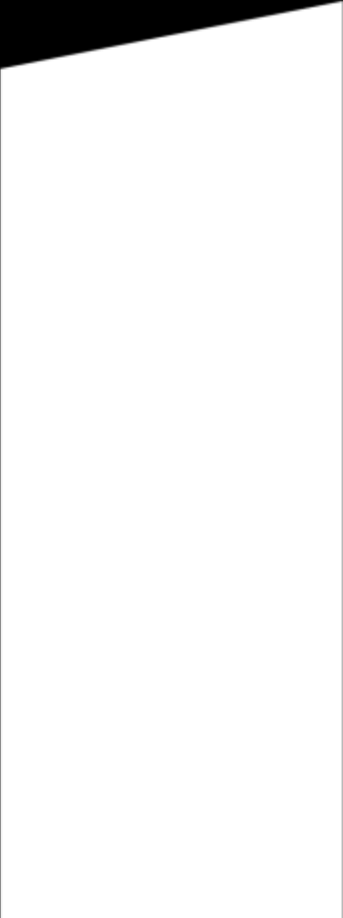


Leaving work after deploying to
production on Friday at 5pm

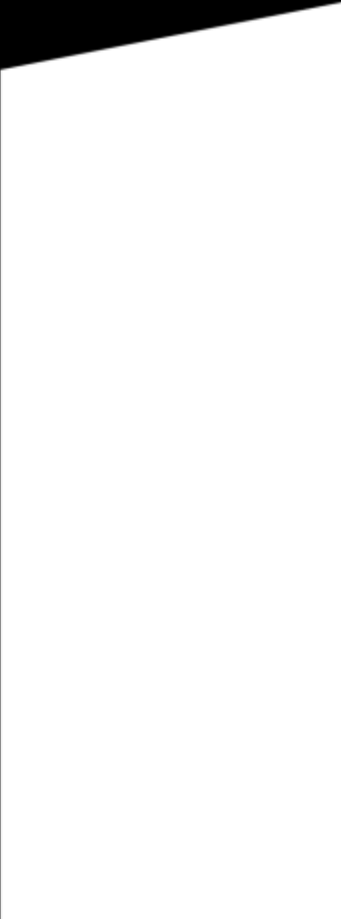
Like a boss

**WORKED FINE IN
DEV**

OPS PROBLEM NOW



**Inn kemur Agile og
bjargvætturinn...
DevOPs!**



**Áður en lengra er
haldið, hver er
ávinningurinn af
DevOps?**

Software delivery performance metric	Low	Medium	High
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	Between once per month and once every 6 months	Between once per week and once per month	On-demand (multiple deploys per day)
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Between one month and six months	Between one week and one month	Between one day and one week
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Between one week and one month	Between one day and one week	Less than one day
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	46%-60%	16%-30%	0%-15%

Elite performers

Comparing the elite group against the low performers, we find that elite performers have...

973x

more frequent
code deployments

6570x

faster lead time
from commit to deploy

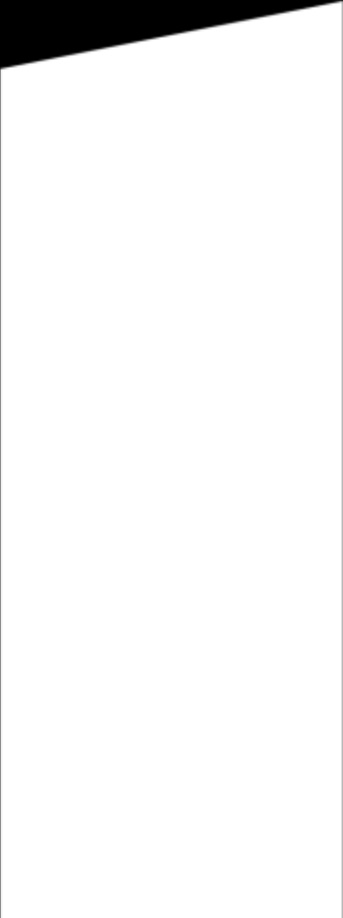
Yes, you read
correctly.
This is not an
editorial error.

3x

lower change failure rate
(changes are $\frac{1}{3}$ less likely to fail)

6570x

faster time to recover
from incidents



Er hægt að mæla?

4 lykilhlutir sem við mælum

1. Deployment Frequency

**2. Mean Lead Time for
Changes**

3. Change Failure Rate

4. Time to Restore Service

1. Deployment Frequency

Hérna erum við að mæla hversu oft við gefum út kóða í raun umhverfin okkar eða gefum út hugbúnað til enda notenda. Eitt af markmiðum DevOps er continuous development sem ýtir teymum í að komast á þann stað að getað gefið út nokkrum sinnum á dag þess vegna. Þessi mælieining gefur okkur góða mynd af því hvar við stöndum.

2. Mean Lead Time for Change

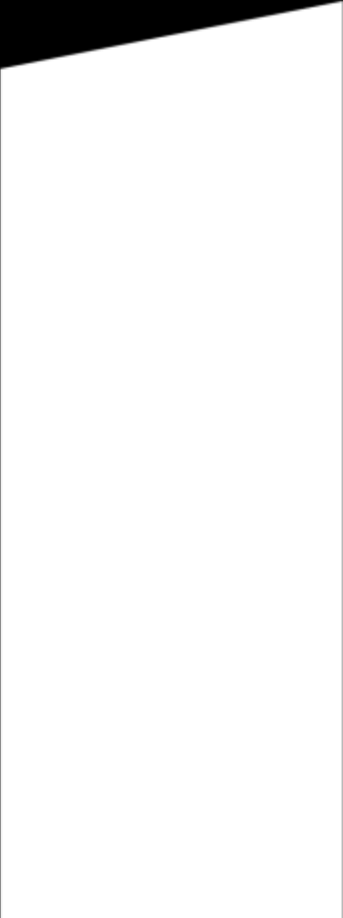
Hérna erum við að mæla meðaltal þess tíma frá því að við tékkum kóða inn og þangað til að kóði er kominn inn í raunumhverfi. Þetta er mikilvæg mælieining, þar sem að því styttri sem þessi tími er, því fyrr fá teymin okkar endurgjöf og því hraðar gefa þau út hugbúnaðinn okkar.

3. Change Failure Rate

Hér erum við að skoða prósentutölu af útgáfum sem valda truflunum í raunumhverfunum okkar, sem þarfnast lagfæringar strax, þ.e. eitthvað sem veldur niðritíma eða röskun á þjónustu. Það segir sig sjálft að hér viljum við sjá töluna sem lægsta, þar sem að ef að teymin okkar eru að eyða mikið af tíma sínum að eltast við lagfæringar eftir að við gefum út, að þá geta þau ekki verið að vinna að nýjum lausnum.

4. Time to Restore Service

Þessi mælieining hefur einnig stundum verið kölluð Mean Time to Recover eða Mean Time to Repair. En þetta mælir hversu fljótt teymin okkar geta komið þjónustu í eðlilegt ástand eftir atvik. Atvik getur verið þá allt frá því að vera galli í kóða yfir í óskipulagðan niðritíma.



**En hvernig höfum
við gert þetta?**

**Dev
Teymi**



**Ops
Teymi**



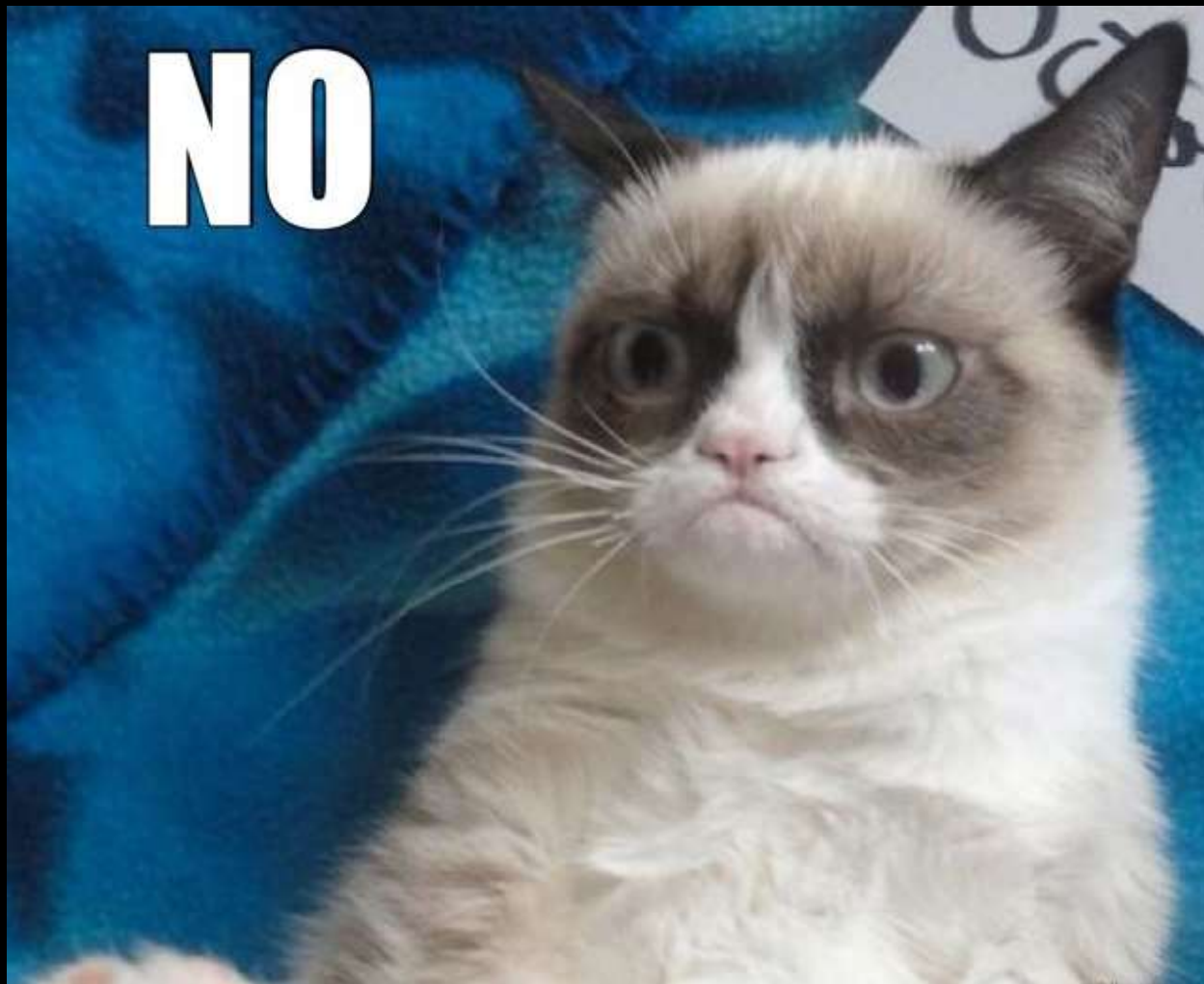
**Dev
Teymi**

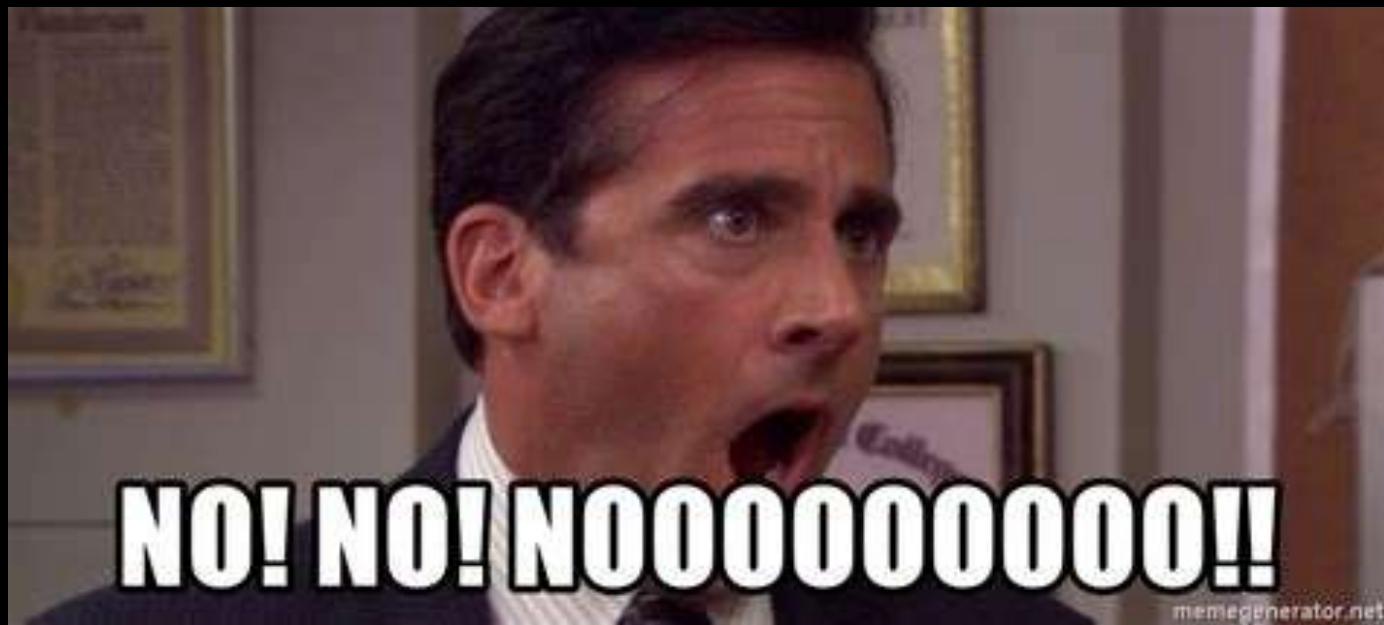


DevOpsTeymi



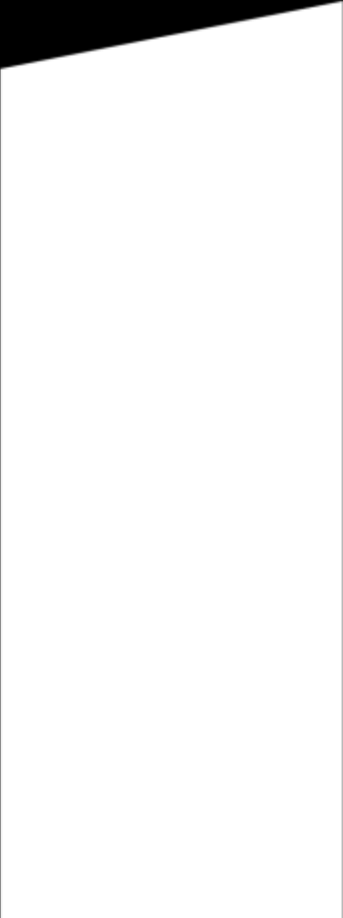
NO



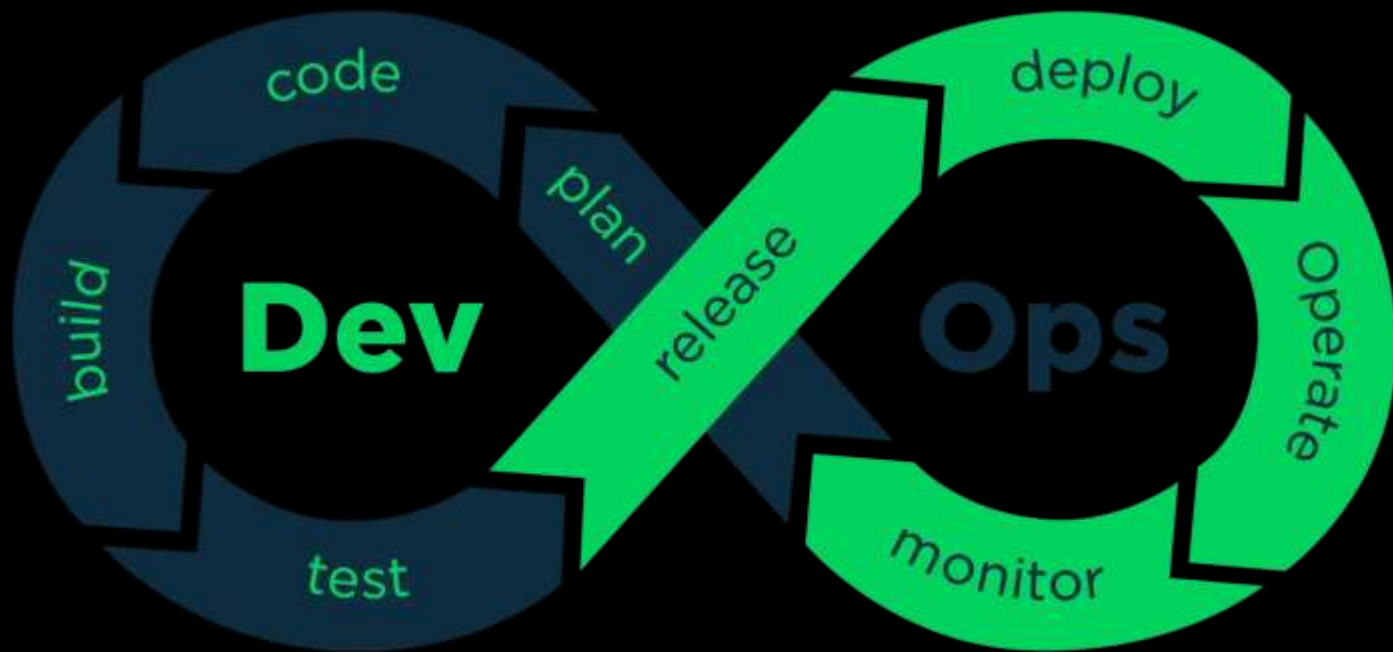




**DevOps er ekki
hlutverk, það er
kúltúr.**

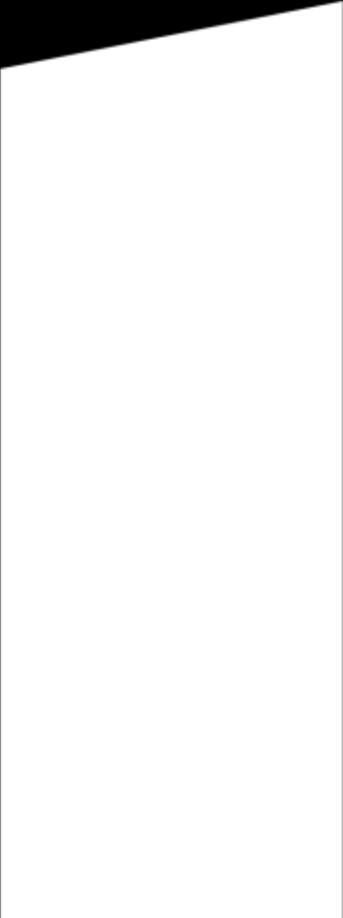
A white geometric shape, resembling a parallelogram or a trapezoid, is positioned on the left side of the slide. It has a slanted top edge and a vertical right edge.

**DevOps is the union of
people, process and
technology to
continually provide
value to our customers**



EN HVAD ER PÁ

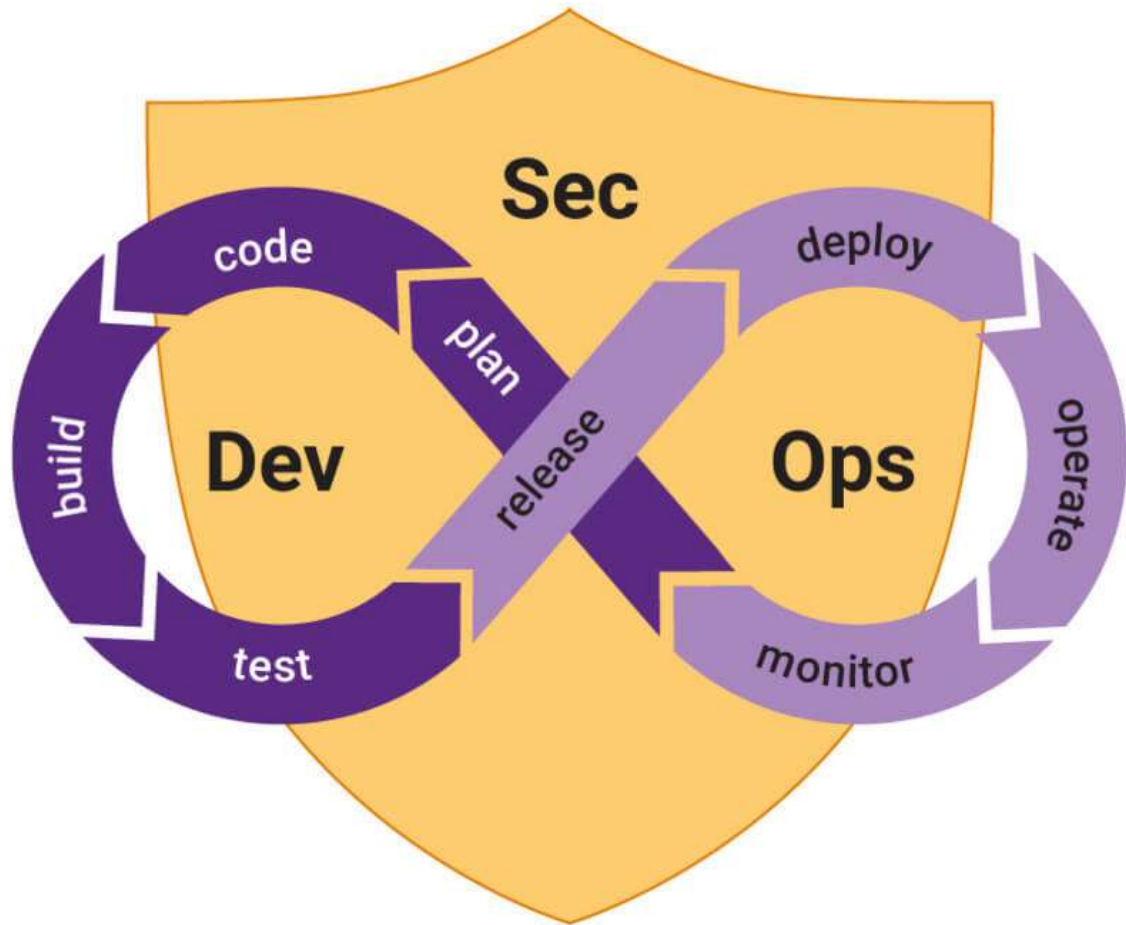
DEVSECOPS?

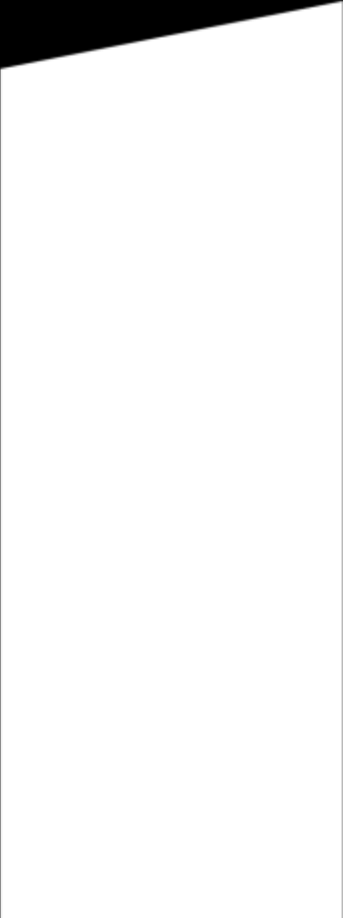
A white geometric shape, resembling a parallelogram or a tilted rectangle, is positioned on the left side of the slide.

Kúltur, alveg eins og DevOps

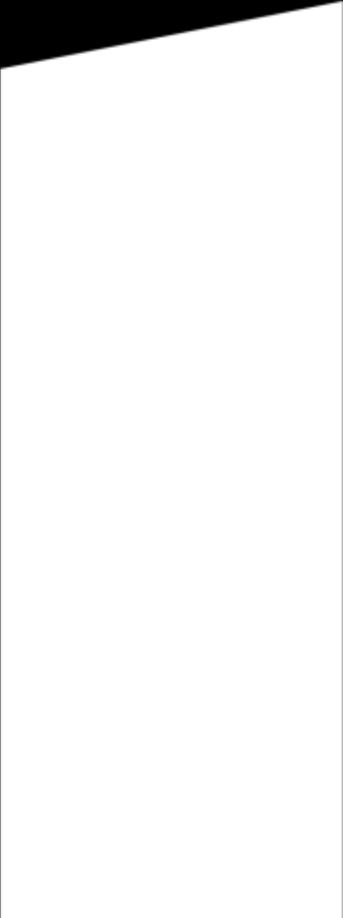


DevOps vs Security

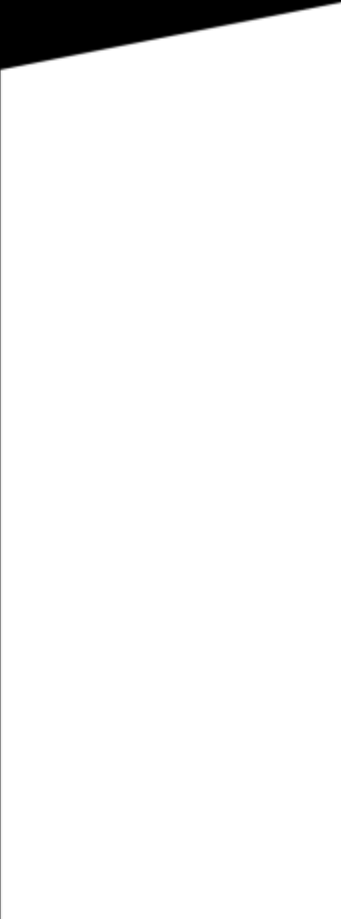


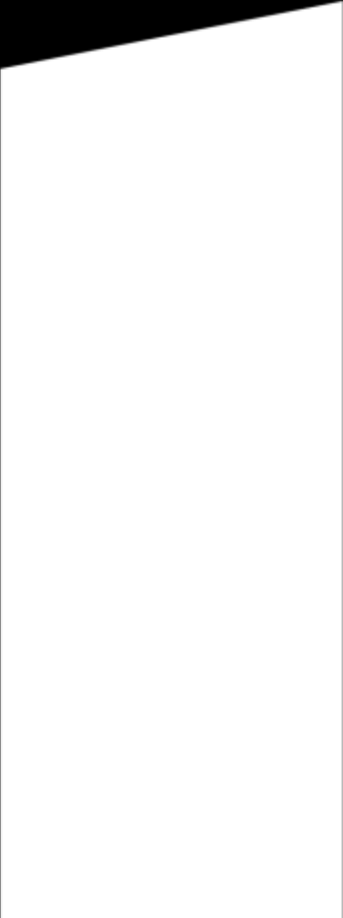


En hvernig...?

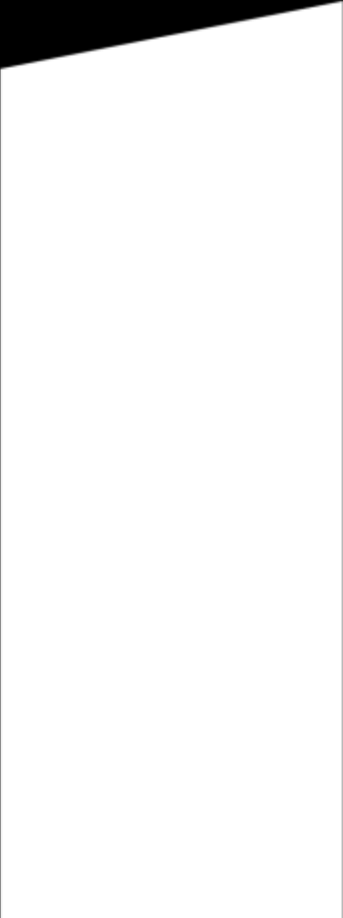


**Byggjum öryggi inn í
hugbúnaðargerðina
okkar**

- 
1. Skilgreinum “shift left” öryggisstefnur
 2. Tökum sjálfvirknivæðingu í öryggistólum fagnandi
 3. Lögum öryggisgalla um leið og þeir koma fram
 4. Fellum sýnileika inn í kúltúrinn okkar



Platform Engineering



Platform engineering is the discipline of designing and building toolchains and workflows that enable self-service capabilities for software engineering organizations in the cloud-native era. Platform engineers provide an integrated product most often referred to as an “Internal Developer Platform” covering the operational necessities of the entire lifecycle of an application.

End-user app

Dev
Teymi



Security

Dev
Teymi



Automation

Dev
Teymi



IaC

Dev
Teymi



End-user app

Security

Dev
Teymi



Dev
Teymi



Platform Engineering

Dev
Teymi



Automation

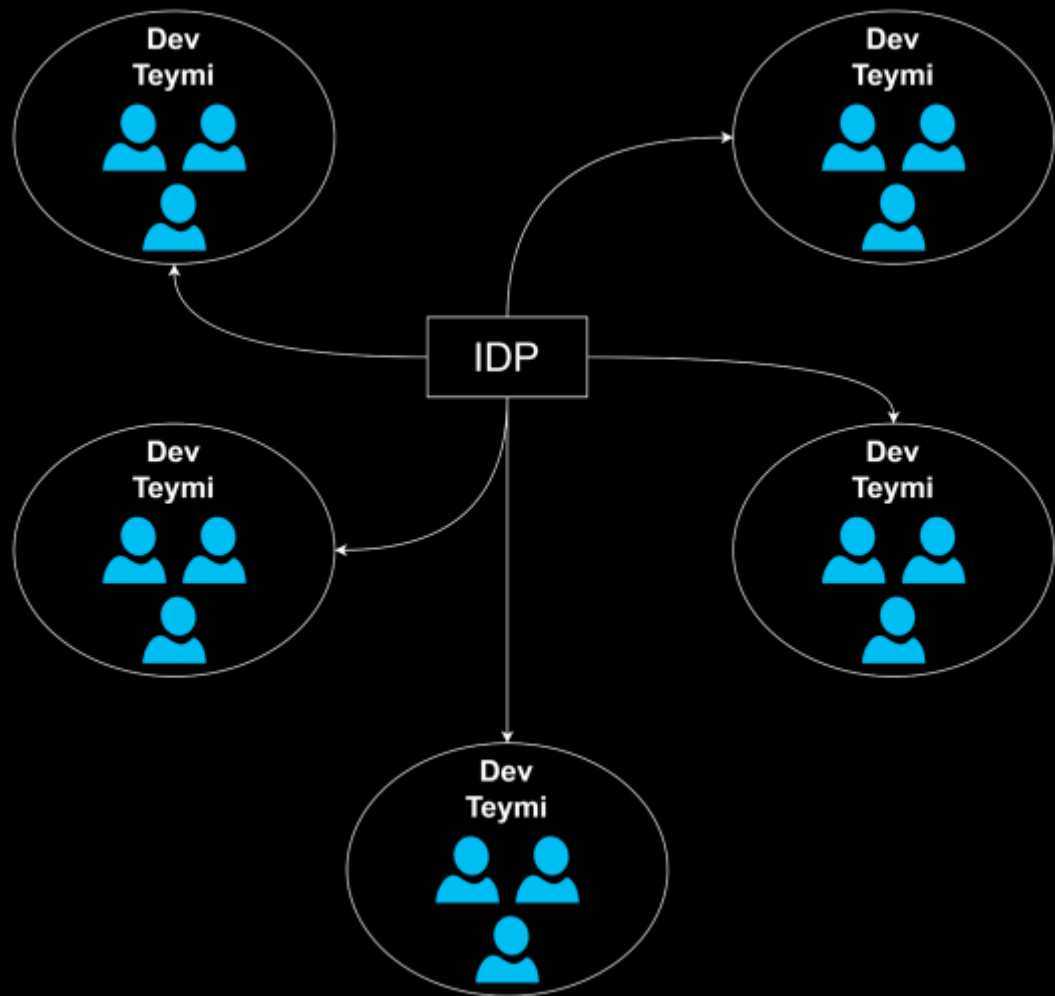
Dev
Teymi

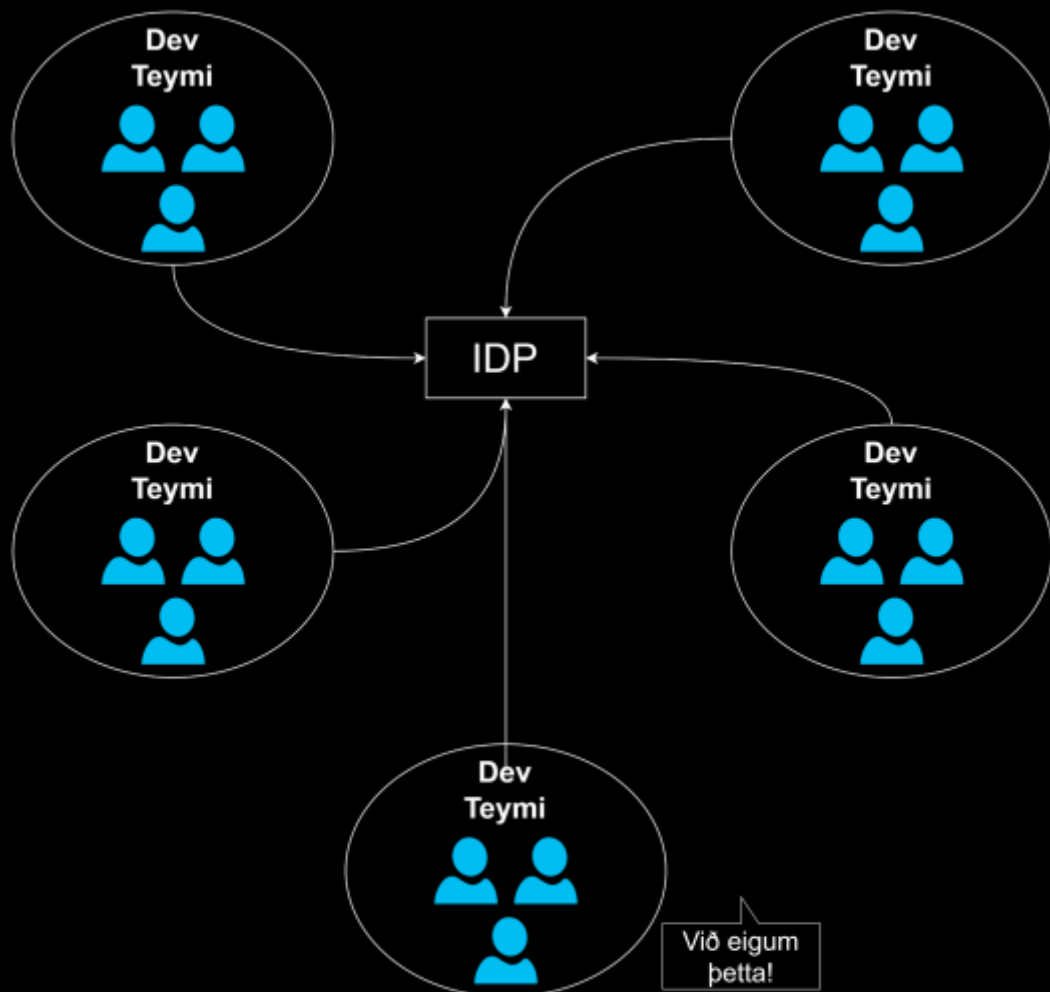


IaC

Dev
Teymi

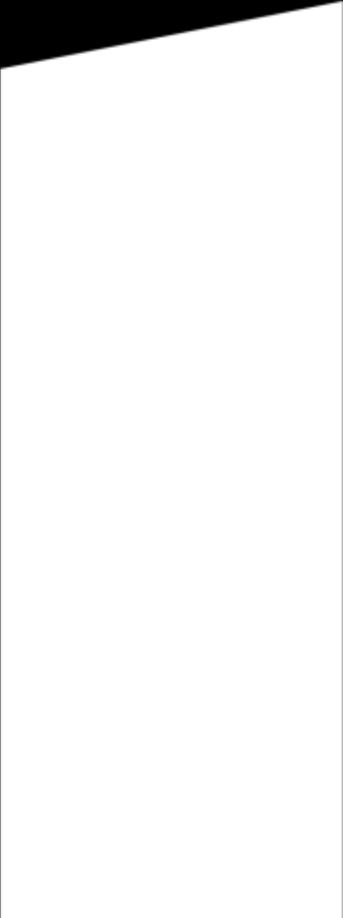






Við eigum þetta!

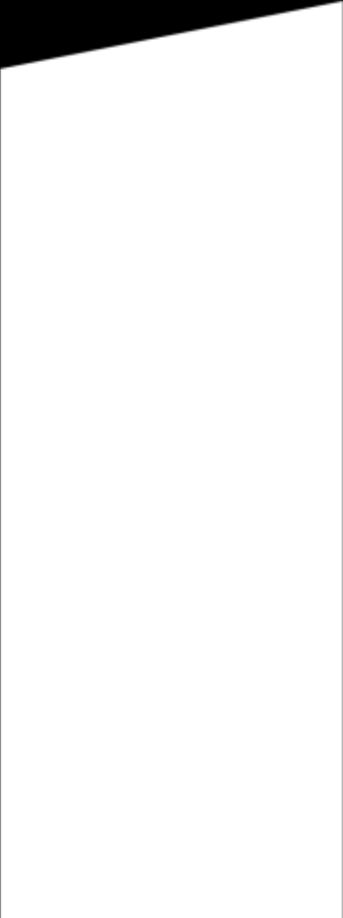
Platform Engineering



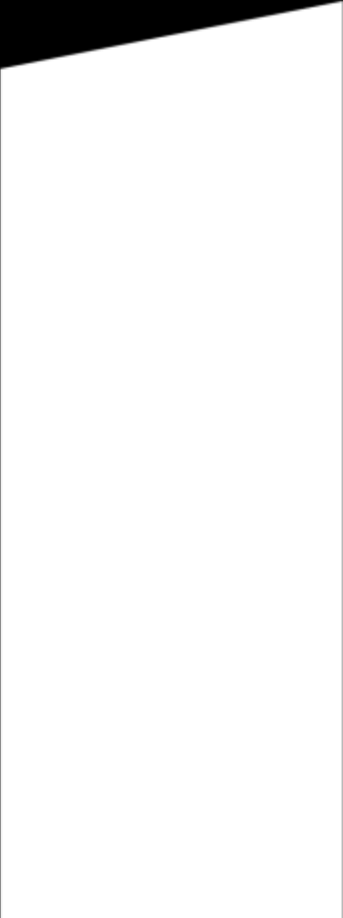
**Hvernig tengist
SRE inn í þetta og
hvaðan kemur
það?**

SRE byrjaði hjá Google





**En ég er ekki FB,
Google, Netflix og
solleiðis, hvað kemur
þetta mér við?**



Site reliability engineering (SRE) is the practice of using software tools to automate IT infrastructure tasks such as system management and application monitoring. Organizations use SRE to ensure their software applications remain reliable amidst frequent updates from development teams. SRE especially improves the reliability of scalable software systems

SRE Key Practices



Innleiðing á SRE gefur okkur...

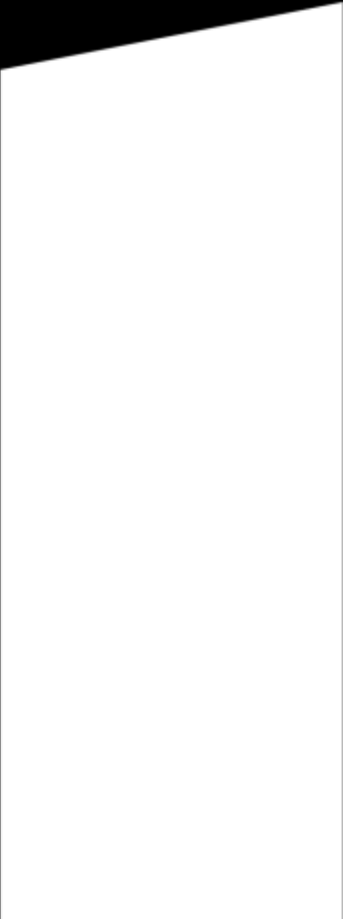
- Styttri “Mean Time to Repair” og “Mean Time between Failures”
- Hraðari útgáfu á uppfærslum og villuleiðréttingum
- Minni líkur á starfsmanna “burnout” þar sem hefðbundnari Ops verkefni verða betrubætingar en ekki slökkvistarf
- Mikla samleiðni á milli hugbúnaðarteyma og SRE teymis þar sem þau hafa sömu markmið
- Aukin öryggisvitund og auðveldar að ná stöðlum

SRE Principles

- Byggja upp CI/CD ferla til að sjálfvirknivæða skölun á infrastrúktúr
- Setja hámark á Ops vinnu: Bara 50% fer í daglegt “vesen”, að minnsta kosti 50% eiga að fara í að betrumbæta kerfin okkar
- Búa til Service Level Agreement (SLA), Service Level Objectives (SLO's) og Service Level Indicators (SLI's) fyrir þjónusturnar okkar
- Setja upp villu budget

SRE Principles

- Koma á laggirnar góðu eftirlitsumhverfi sem fylgist með latency, saturation, traffic og villum.
- Búa til viðbragðsáætlun til að taka á atvikum. Reyna að sjálfvirknivæða þetta
- Halda blameless postmortems



SLI vs SLA vs SLO?

SLO

Service-Level Objective (SLO) er nákvæm tala sem við setjum á tiltækileika kerfanna okkar. Allar framtíðar umræður um það hvort að viðkomandi kerfi sé að keyra eðlilega eða hvort við þurfum að gera einhverjar breytingar á arkitektúr kerfisins þurfa alltaf að vera rammaðar þannig að kerfið haldi áfram að mæta SLO'inu sínu.

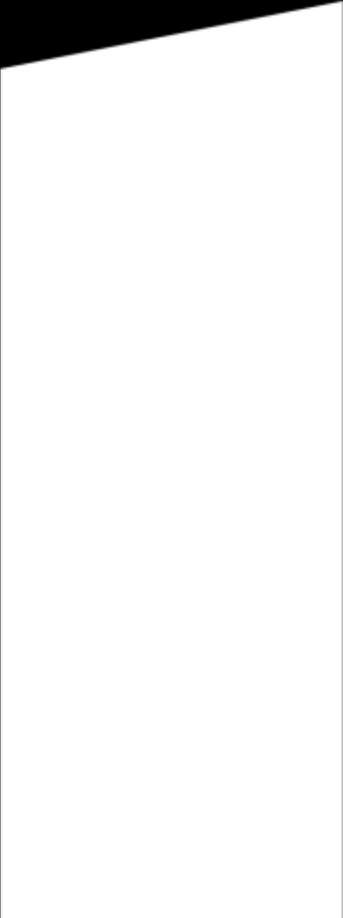
SLA

Service Level Agreement (SLA) er samningur sem við gerum við okkar notendur, hvort sem þeir eru innanhúss eða utanhúss um afleiðingar þess ef að við förum undir SLO'íð okkar.

Oftast koma SLA ekki frá SRE teyllum, heldur frá viðskiptahlið eða vöruhlið fyrirtækisins. SRE hins vegar hjálpar til við að komast hjá því að SLO klikki og hjálpar einnig við að skilgreina SLI

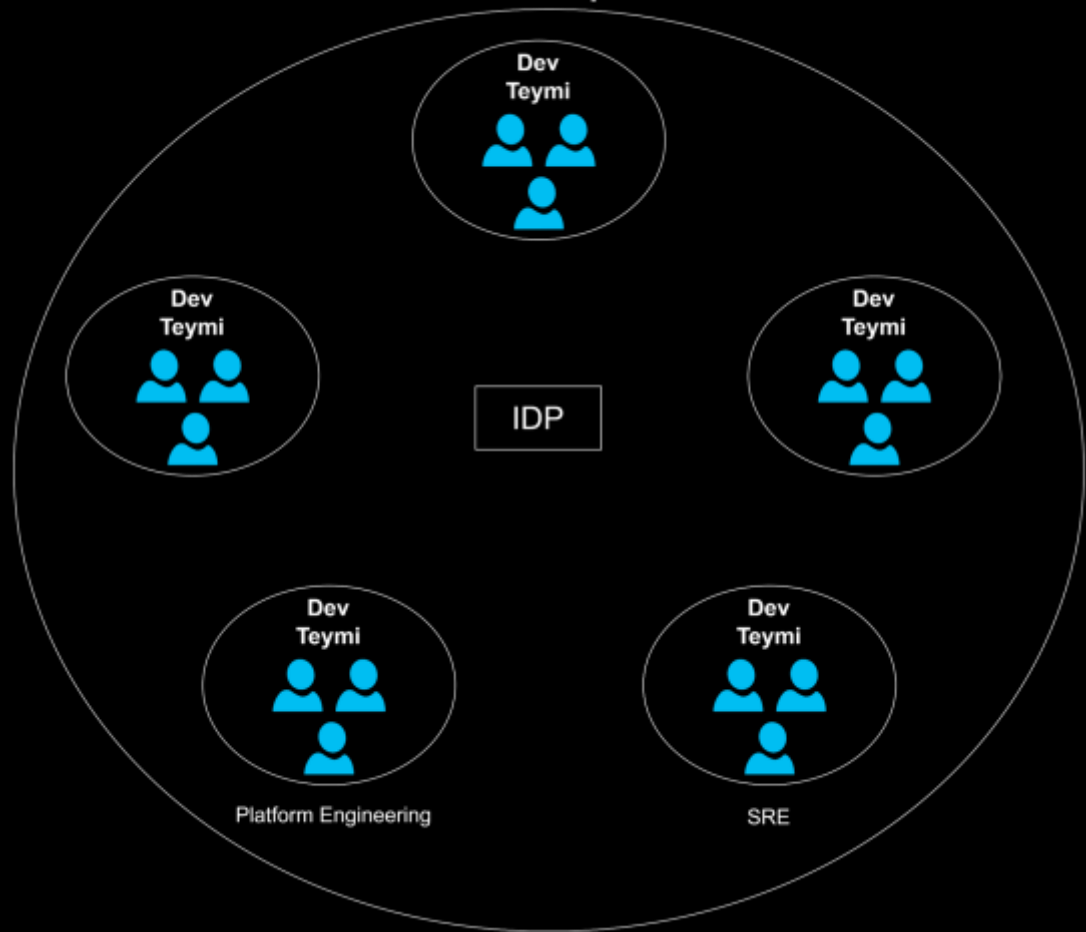
SLI

Service-Level Indicator (SLI) er mælieining um það hvernig þjónustan okkar er að haga sér. Þegar við skoðum hvort að kerfin og þjónusturnar okkar eru búnar að vera að keyra innan SLO yfir einhvern tíma, þá skoðum við SLI til að sjá prósentuna um það hvernig tiltækileikinn er búinn að vera. Ef við viljum vita hversu áreiðanlegar þjónusturnar okkar eru, þá verðum við að geta mælt hvernig kerfi/þjónustur eru að standa sig sem verður okkar SLI



**Hvernig passar
þetta allt saman?**

DevOps



www.themainframe.com