jcon.one

JCON2022

HOW WE BUILD AND MIGRATED

our Spring Boot Applications to Kotlin

JAVAPRO

MicroStream

Gradle Enterprise

XDEV

RAPIDclipse™ X
The visual Eclipse

sessionize

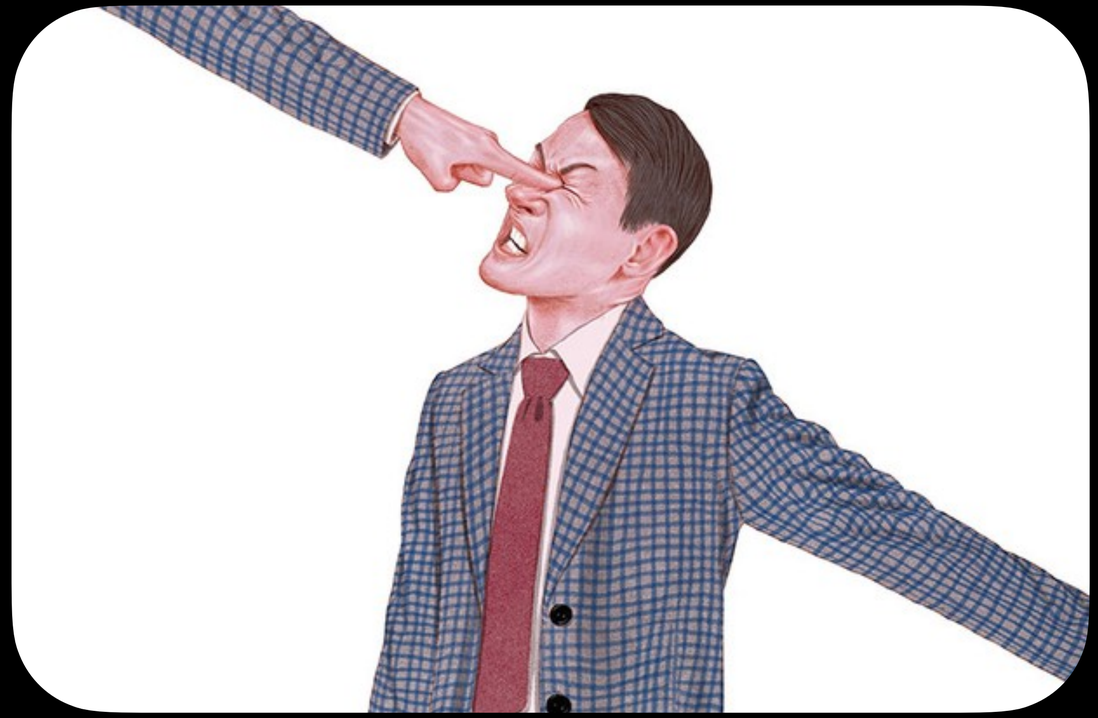
CONRAD

QA|WARE
SOFTWARE ENGINEERING

ECLIPSE FOUNDATION

aws

What's in it for



You will learn

How to:

- Migrate to Kotlin --> IntelliJ IDEA
- Improve your code
  - --> scoped functions
  - --> replace loops
  - --> reactive

WARNING

This meeting will contain live coding,
so everything could (potentially) go wrong.

The story……

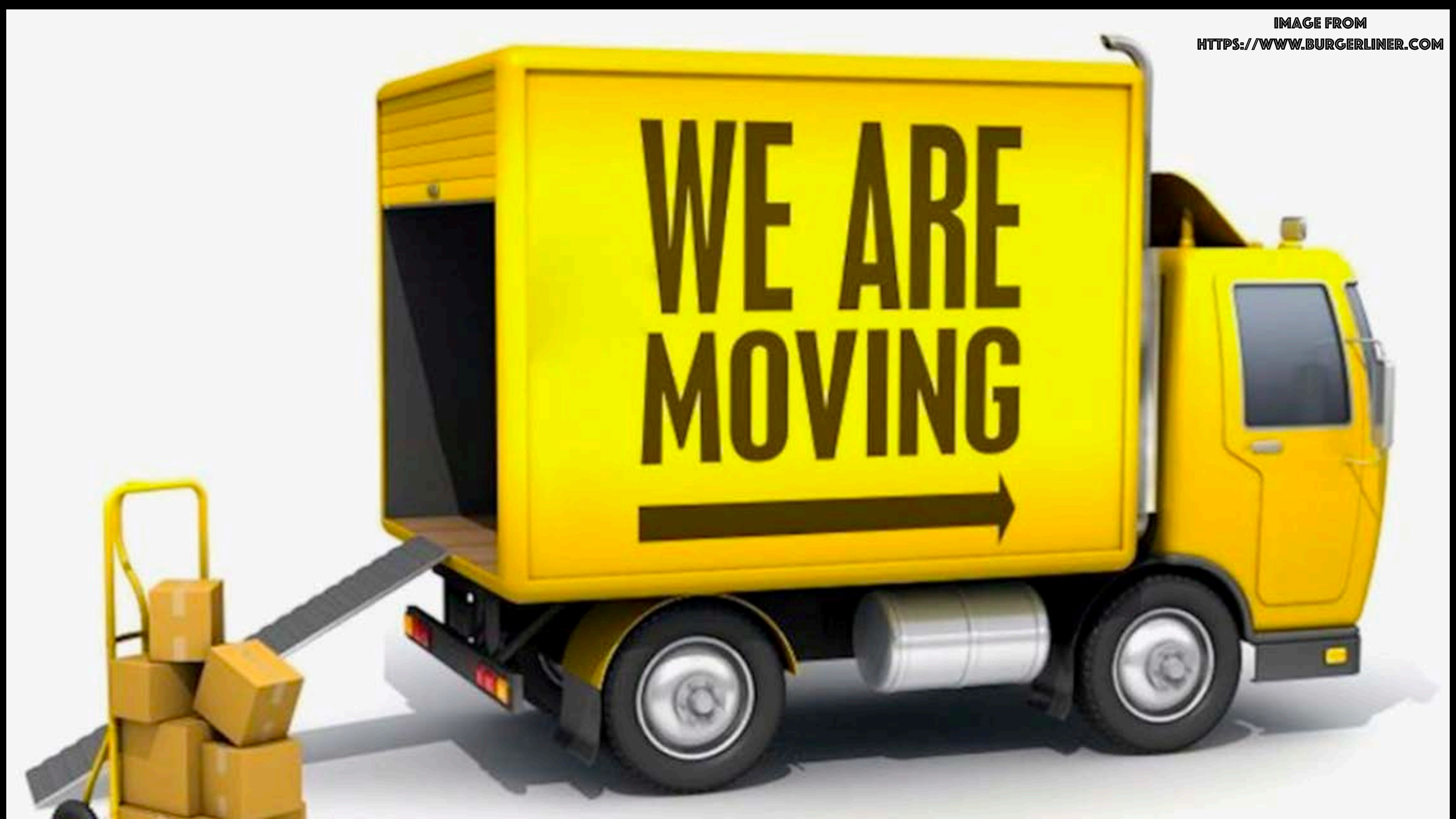


IMAGE FROM
HTTPS://WWW.BURGERLINER.COM

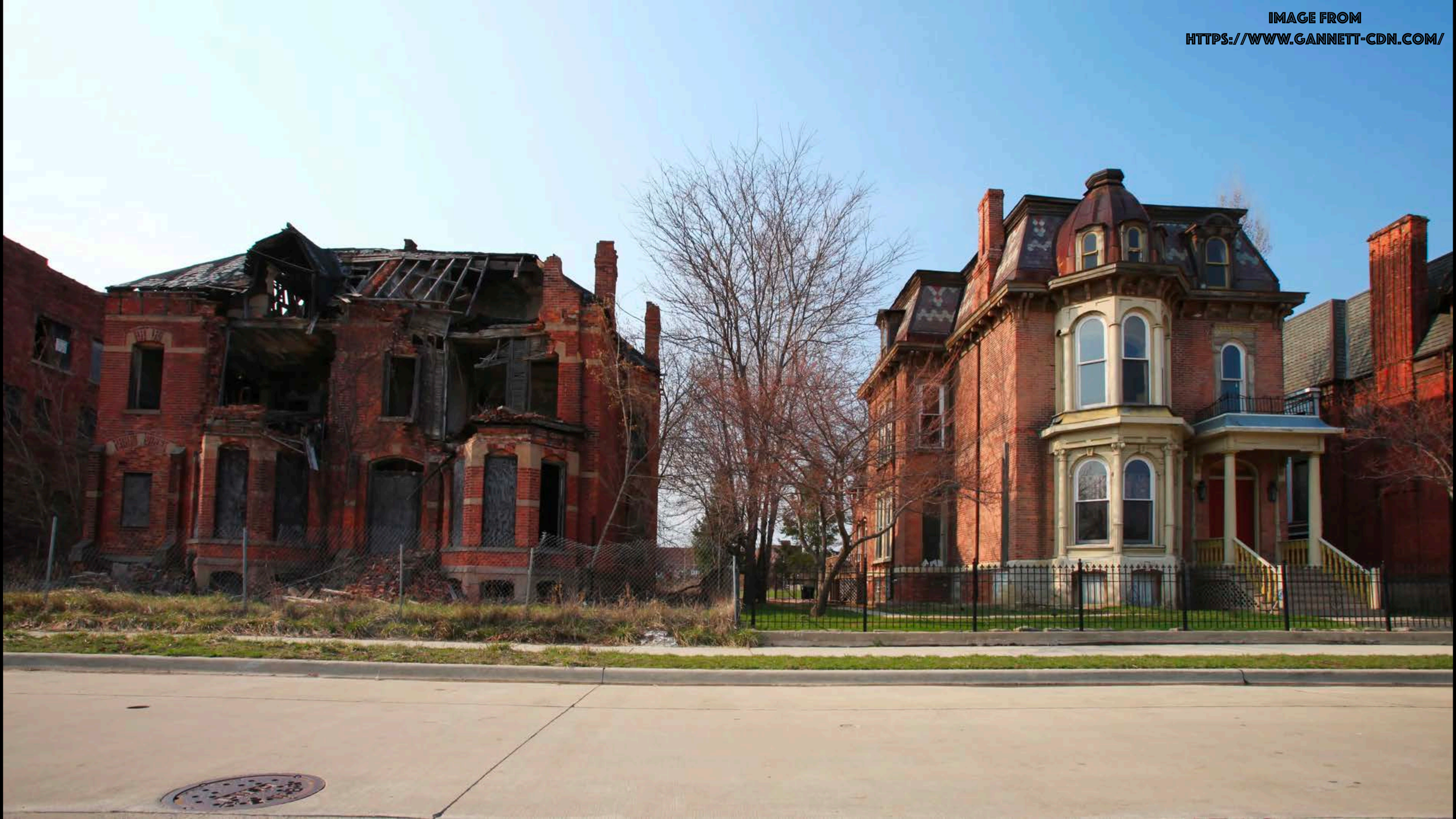


IMAGE FROM
HTTPS://WWW.GANNETT-CDN.COM/

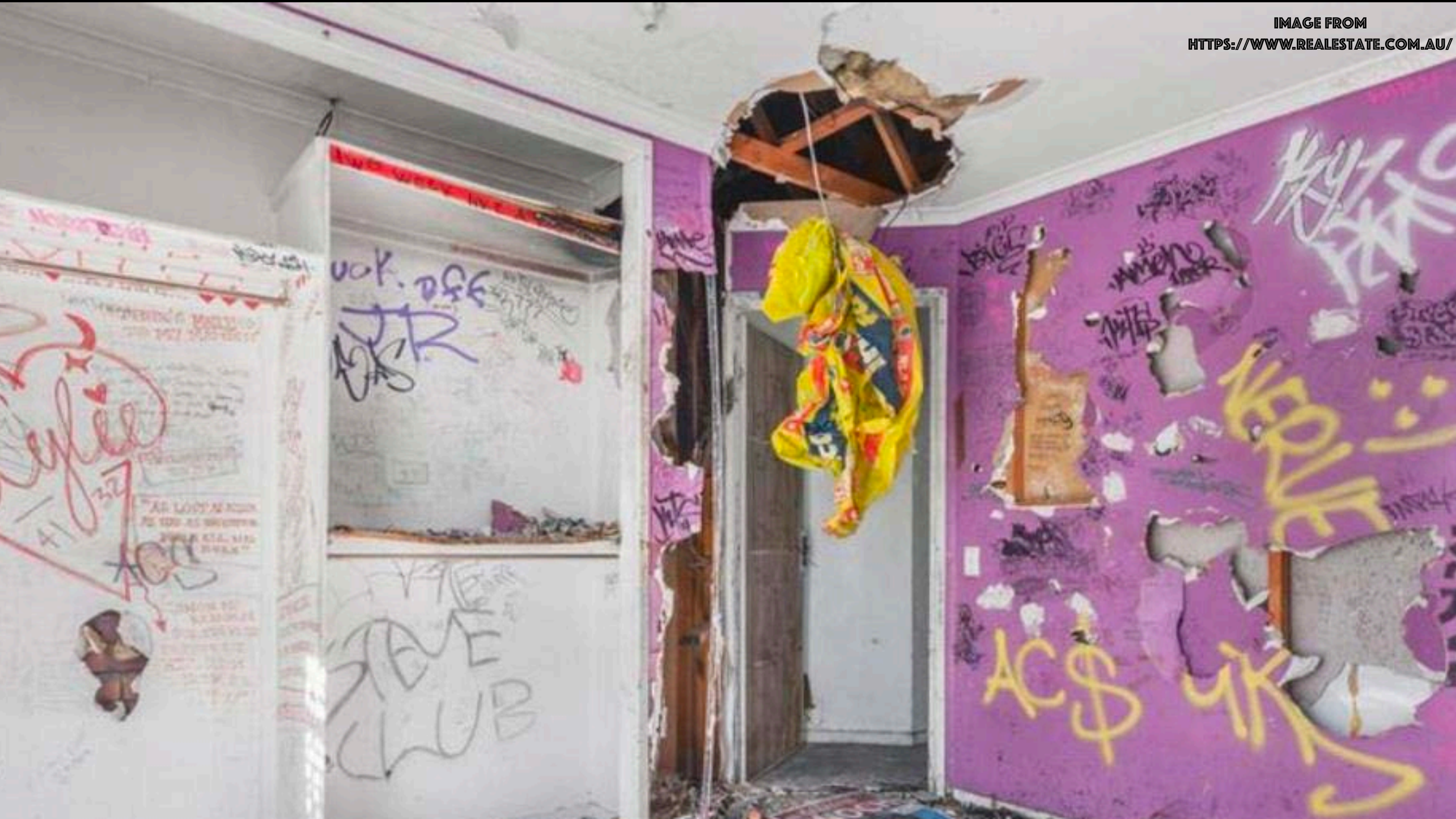


IMAGE FROM
HTTPS://WWW.REALESTATE.COM.AU/

IMAGE FROM
HTTPS://WWW.BROAVLOERISOLATIE.NL/

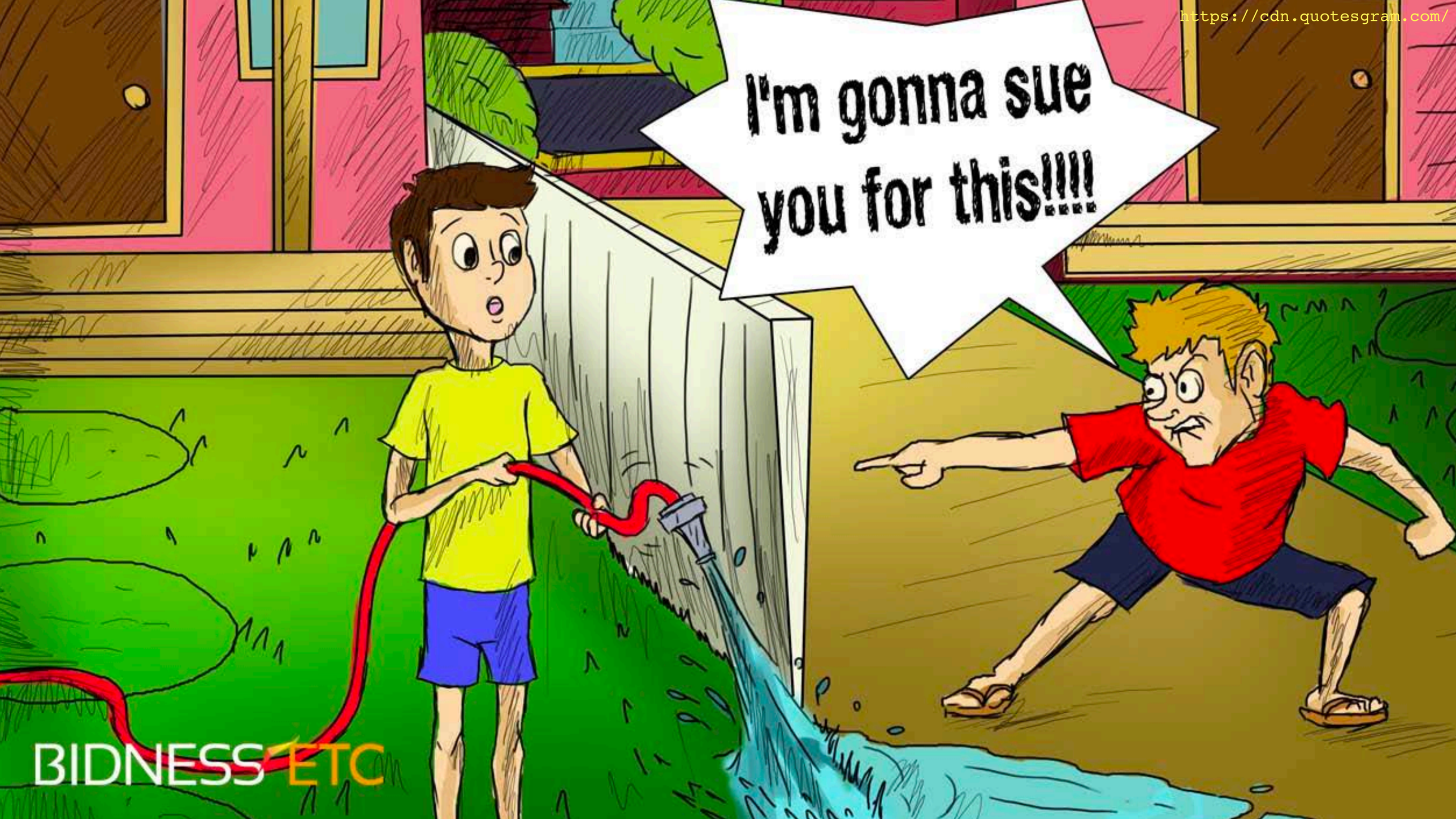


IMAGE FROM
HTTP://METROGUARDSERVICES.CA/

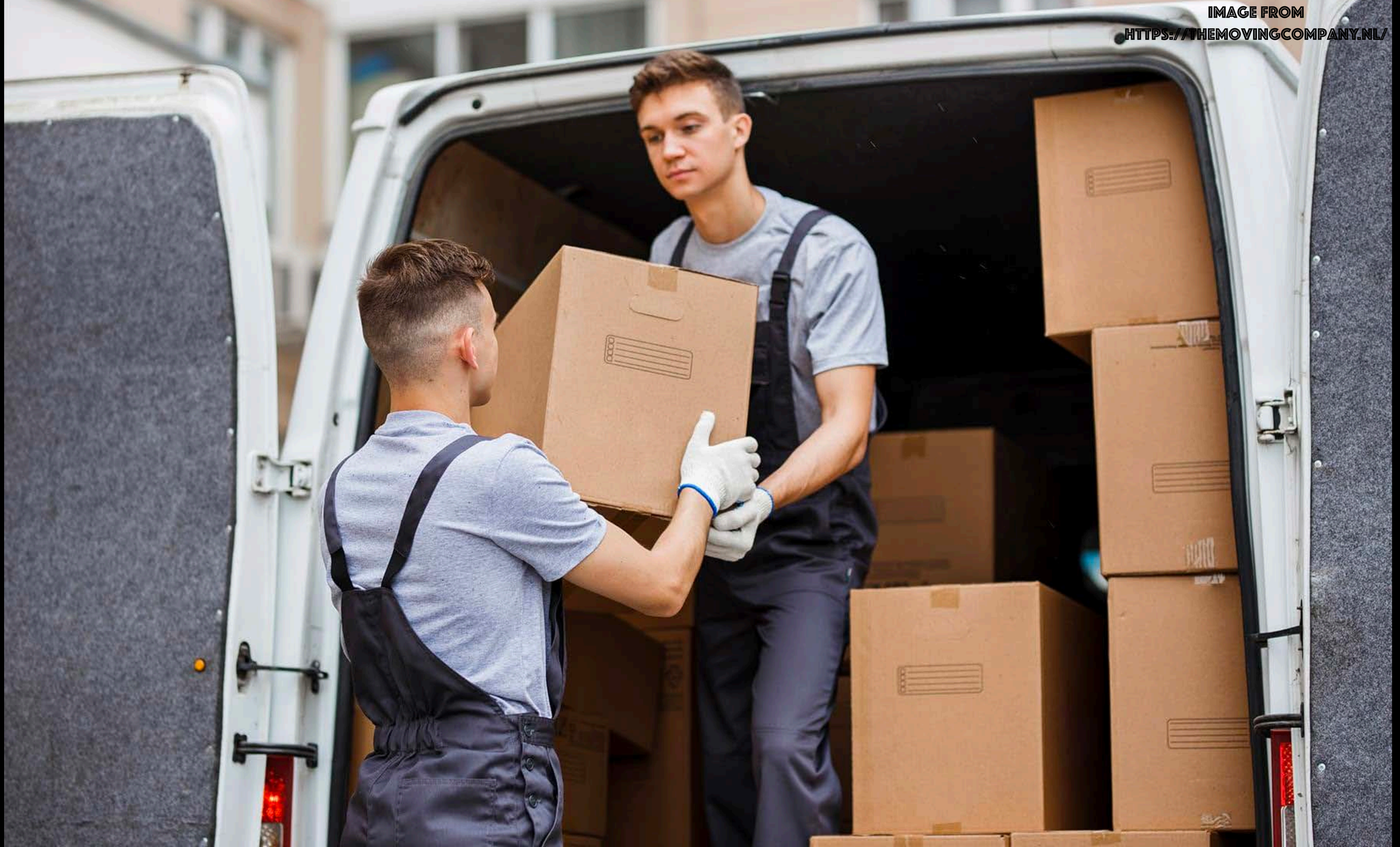


IMAGE FROM
HTTPS://THEMOVINGCOMPANY.NL/

Surprise!

All these steps are also applied when migrating

HOW WE BUILD AND MIGRATED

our Spring Boot Applications to Kotlin

Ko Turk

Developer

blue4IT

Rabobank

https://github.com/KoTurk/Kotlin

I’m telling my experiences

Kotlin ambasador

Speaker

https://twitter.com/KoTurk77

There are different strategies

To move to Kotlin

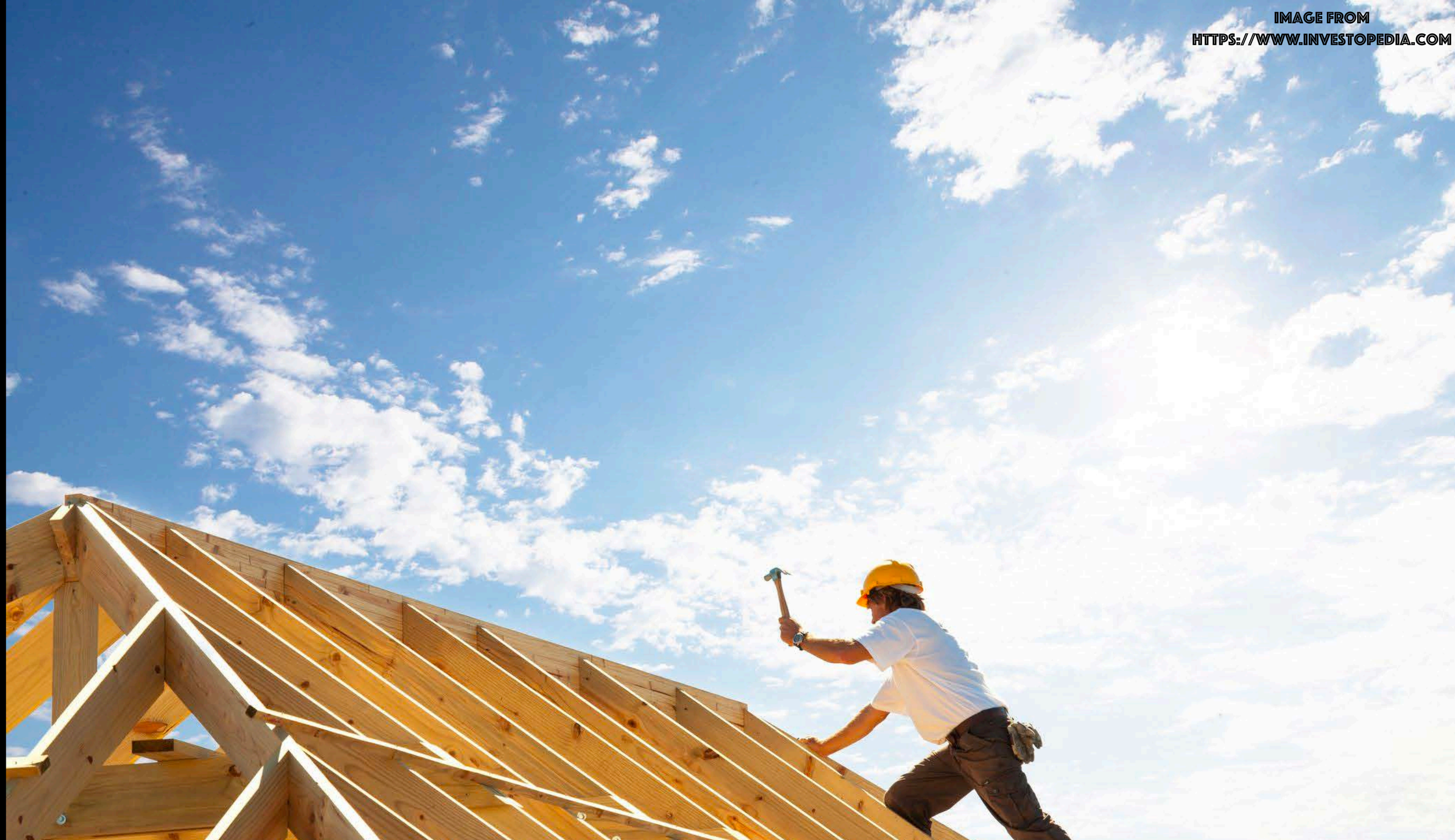


IMAGE FROM
HTTPS://WWW.INVESTOPEDIA.COM

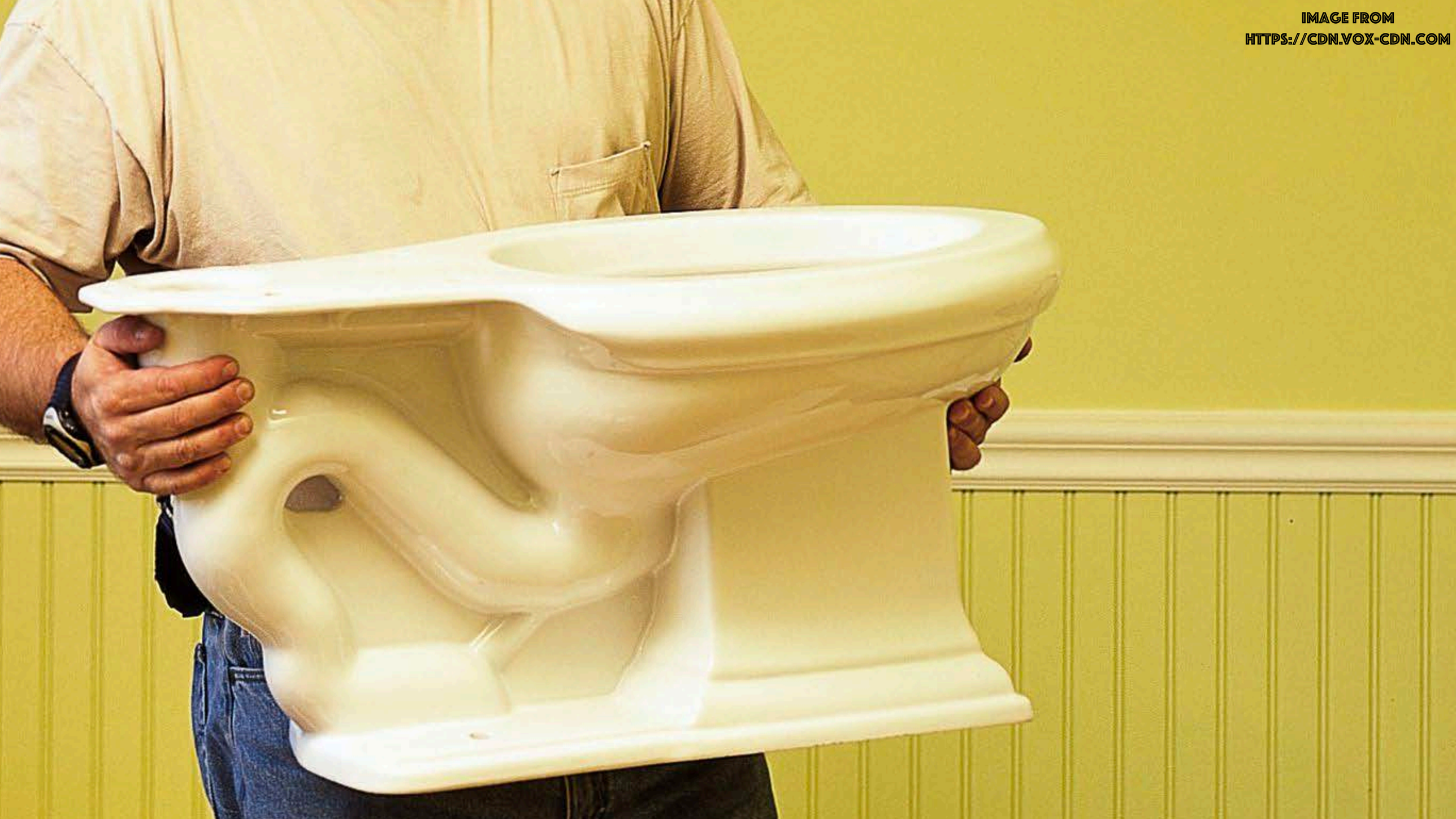


IMAGE FROM
HTTPS://CDN.VOX-CDN.COM

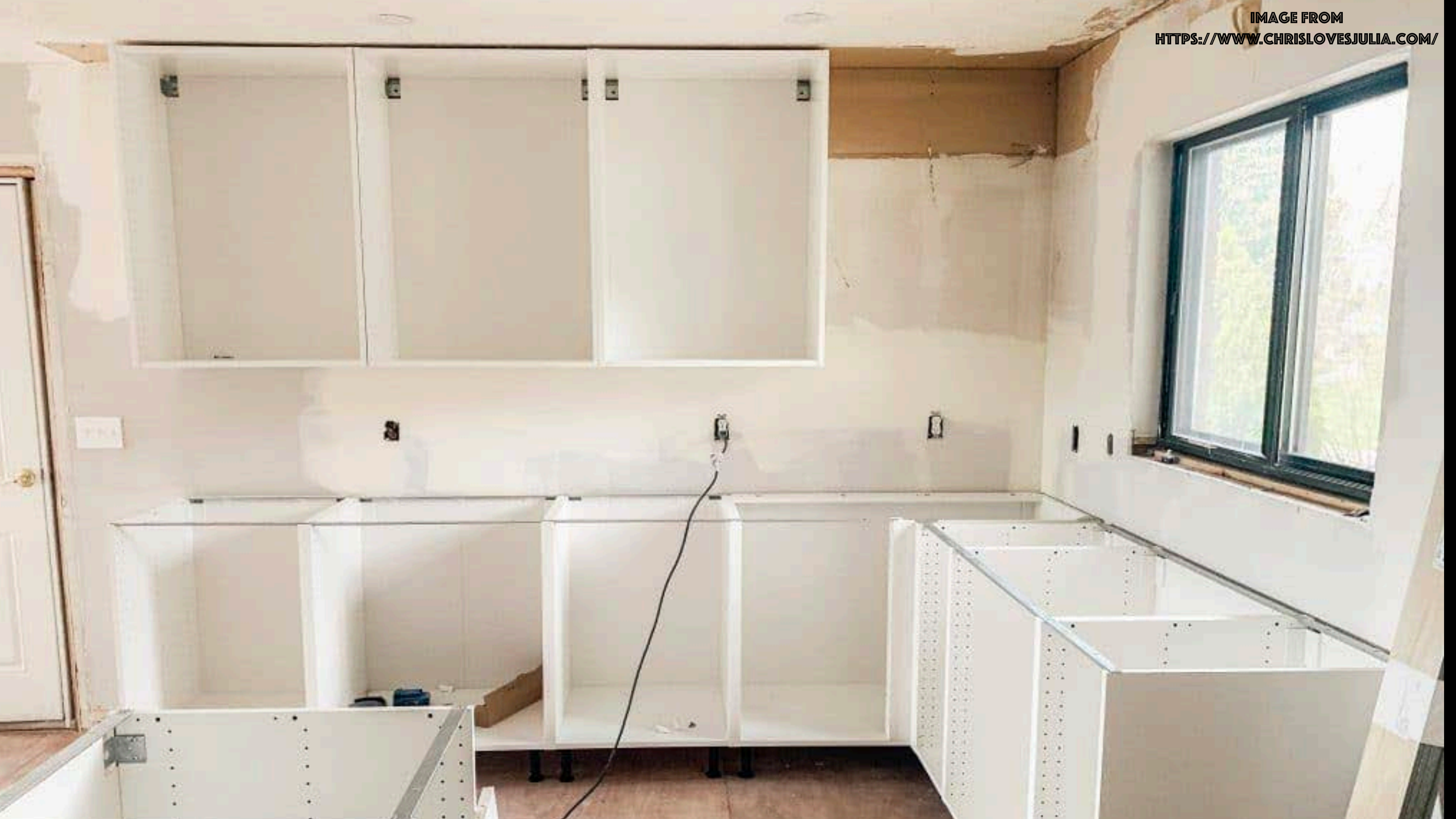


IMAGE FROM
HTTPS://WWW.CHRISLOVESJULIA.COM/

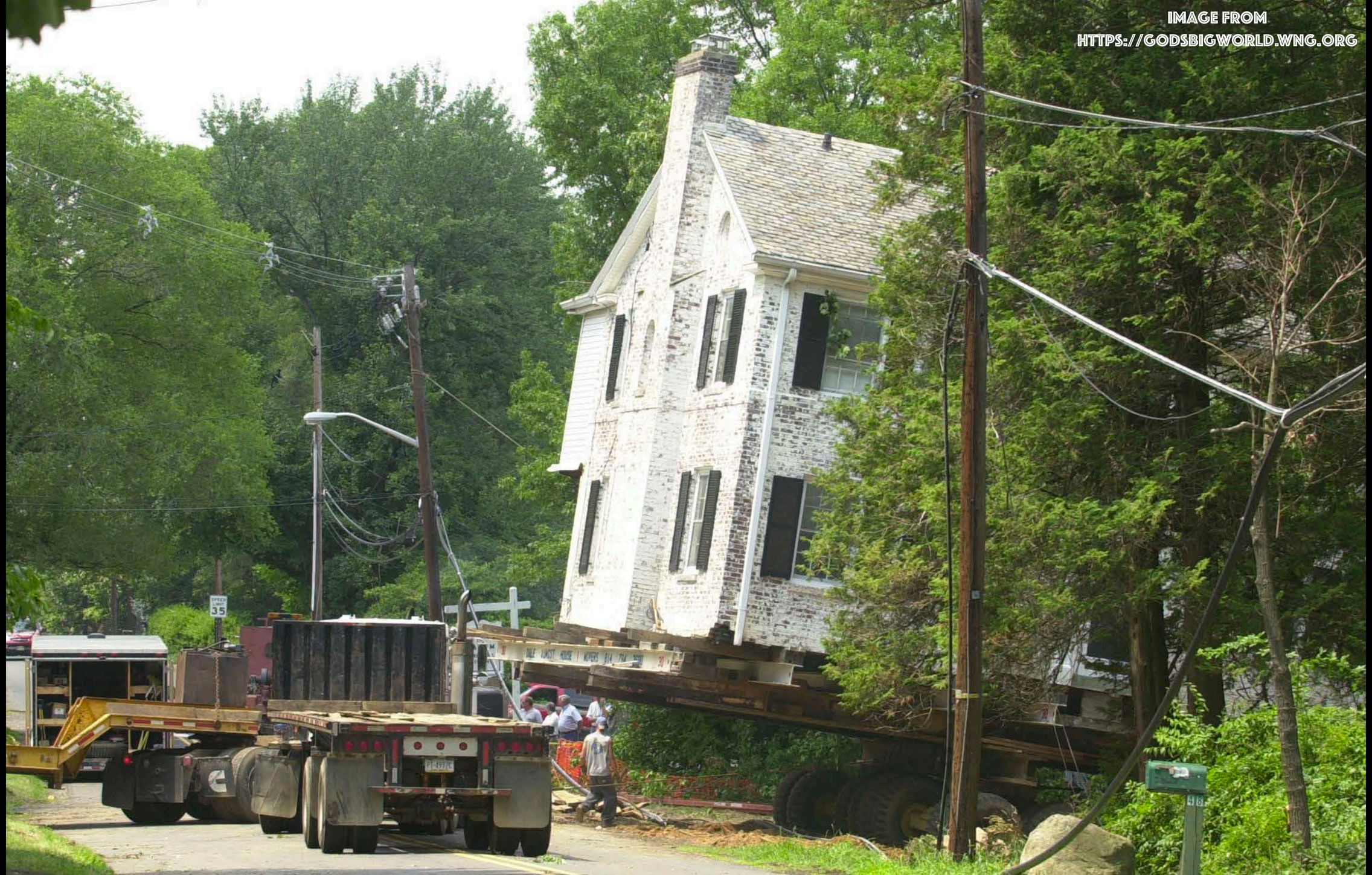


IMAGE FROM
HTTPS://GODSBIGWORLD.WNG.ORG

Live coding

IMAGE FROM
HTTPS://PINNACLEBUSINESSWATER.CO.UK

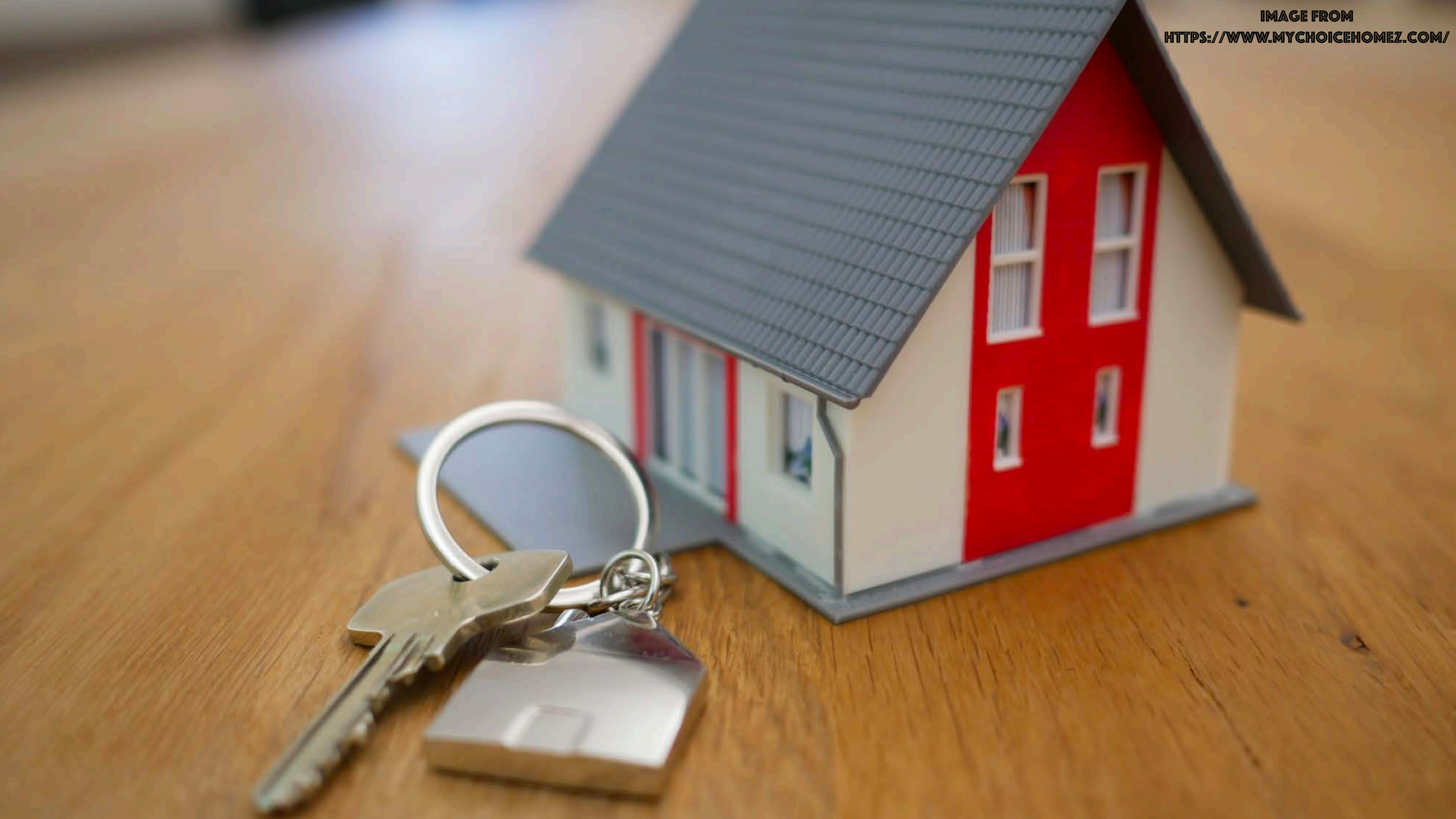


IMAGE FROM
HTTPS://WWW.MYCHOICEHOMEZ.COM/

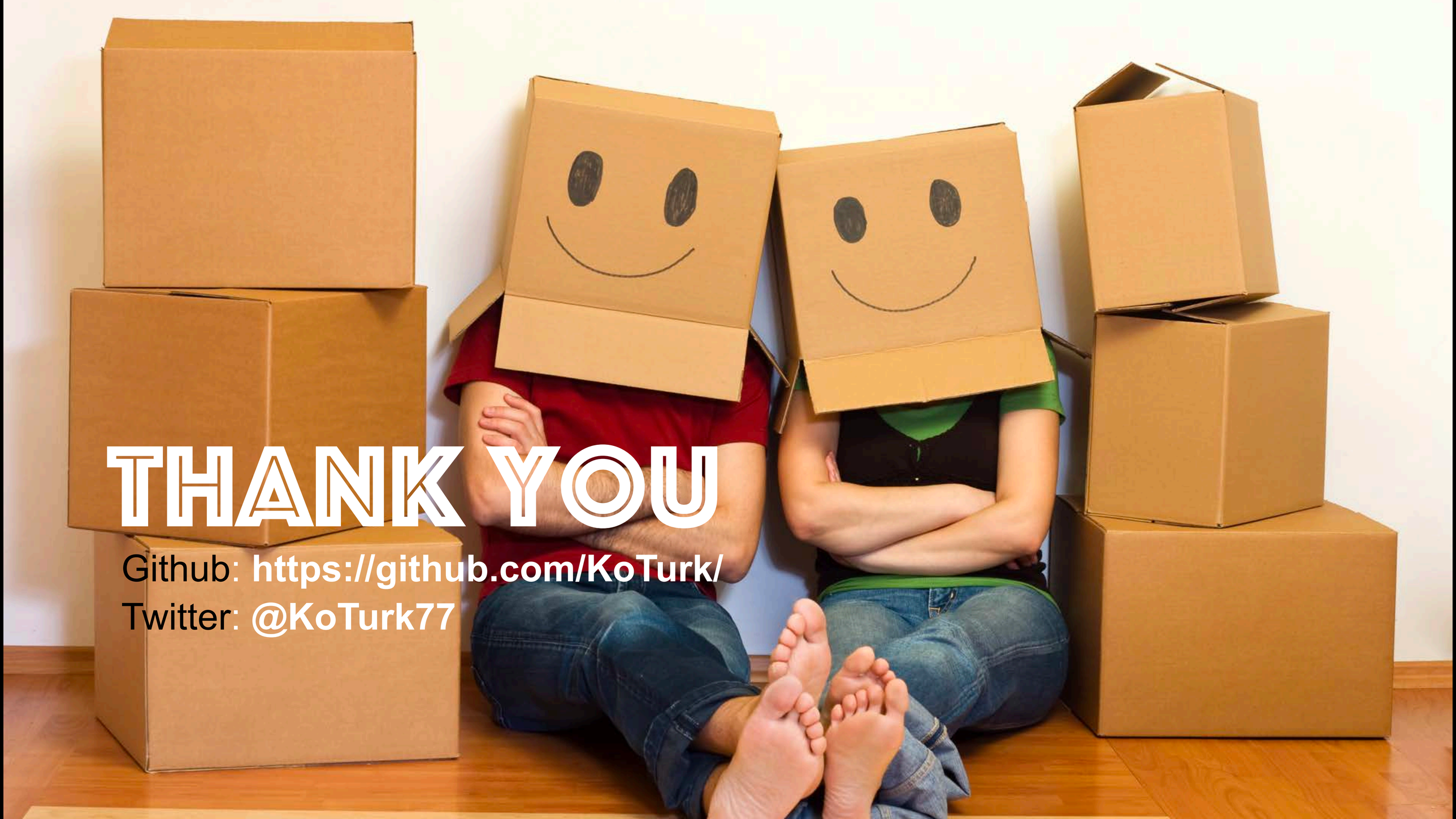
Key takeaways

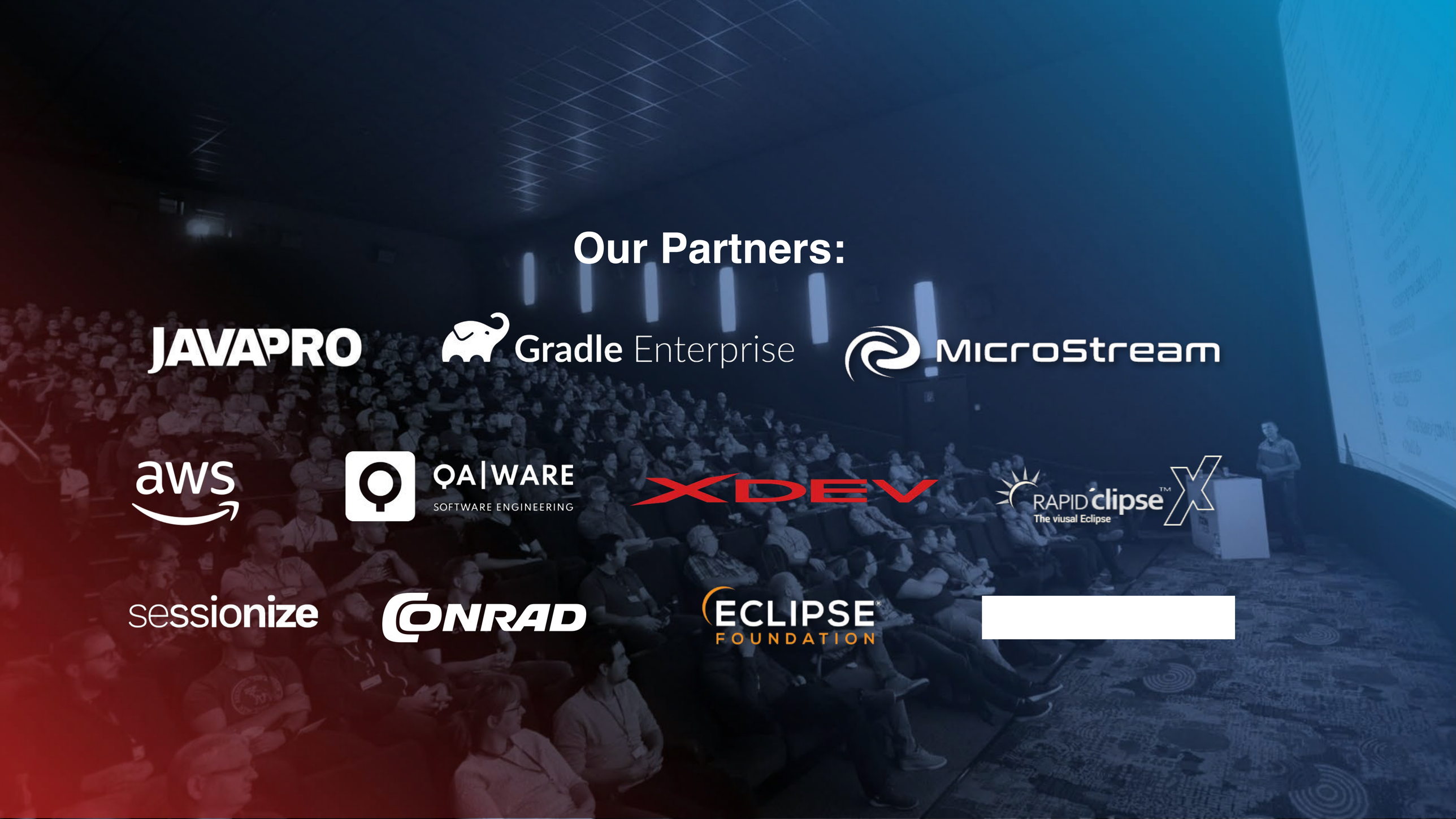
- simplicity **-> migrating is simple!**

- begin simple **-> use data classes, do not start with logic**

- do safe calls **-> with .let, .also and ?:**

- make your code readable **-> with idiomatic Kotlin!**


THANK YOU
Github: https://github.com/KoTurk/
Twitter: @KoTurk77


Our Partners:
JAVAPRO
Gradle Enterprise
MicroStream
aws
QA|WARE
SOFTWARE ENGINEERING
XDEV
RAPID clipse X
The viusal Eclipse
sessionize
CONRAD
ECLIPSE
FOUNDATION

ABOUT COROUTINES

Coroutines aren't a snap-in replacement for threads. The fundamental difference is in the kind of API you're using. The rule of thumb is this:

- blocking API -> use a thread pool
- non-blocking (async) API -> use coroutines

So, if you can get a hold of an asynchronous mail-sending API, then by all means use coroutines with it. But if you're stuck with a blocking API, coroutines won't bring you much value. They can make it a bit more convenient to transfer a blocking operation out of the UI thread, but the mechanics will be the same with or without coroutines.