

# Onboarding Zig at Uber

Motiejus Jakštys

2022-04-10

The Uber logo, consisting of the word "Uber" in a bold, black, sans-serif typeface.

## About me

- ▶ Motiejus Jakštys.
- ▶ At Uber since 2016.
- ▶ Past: Erlang developer, Linux/FreeBSD sysadmin.
- ▶ Cartographer.

# About Uber's tech stack

- ▶ About the company.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.
  - ▶ Lots of cool/innovative tech.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.
  - ▶ Lots of cool/innovative tech.
- ▶ A few thousand engineers.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.
  - ▶ Lots of cool/innovative tech.
- ▶ A few thousand engineers.
- ▶ Go and Java.



# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.
  - ▶ Lots of cool/innovative tech.
- ▶ A few thousand engineers.
- ▶ Go and Java.
  - ▶ Company-wide mandate.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.
  - ▶ Lots of cool/innovative tech.
- ▶ A few thousand engineers.
- ▶ Go and Java.
  - ▶ Company-wide mandate.
  - ▶ Rare exceptions exist.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.
  - ▶ Lots of cool/innovative tech.
- ▶ A few thousand engineers.
- ▶ Go and Java.
  - ▶ Company-wide mandate.
  - ▶ Rare exceptions exist.
  - ▶ Monorepo per language.

# About Uber's tech stack

- ▶ About the company.
  - ▶ Started in 2010.
  - ▶ > 15B trips.
  - ▶ Lots of cool/innovative tech.
- ▶ A few thousand engineers.
- ▶ Go and Java.
  - ▶ Company-wide mandate.
  - ▶ Rare exceptions exist.
  - ▶ Monorepo per language.
- ▶ Go monorepo is about an order of magnitude larger than Linux kernel.

How does Uber use Zig?

# How does Uber use Zig?


Message in  eng



**ctchenn** 

Apr 5th at 18:13



<https://twitter.com/croloris/status/1511342711764303890> who is the speaker can you please join #zig 



**Loris Cro**  | @croloris

Just received confirmation for a talk for the upcoming Zig meetup that might interest some of you:

"How Zig is used at Uber"

## How does Uber use Zig?



**abg** 18:17

[@ctchenn](#) my understanding is that we're using Zig's C toolchain only, not the language. It's not fully rolled out yet, but among other things, it'll enable cross compilation of C-based code (as well as Go code that uses CGo). It'll drop the dependency on the system's C compiler.





## About this talk.

- ▶ Why we need Zig at Uber.

## About this talk.

- ▶ Why we need Zig at Uber.
- ▶ How we onboarded it.

## About this talk.

- ▶ Why we need Zig at Uber.
- ▶ How we onboarded it.
- ▶ How are we going to use it.

# Timeline

## 2018: Monorepo

Go services start moving to The Go Monorepo en masse.

# 2019: Hermetic C++ toolchain



Go Monorepo / GM-221

## Add deterministic cxx toolchain to go-code

[Edit](#) [Add comment](#) [Assign](#) [More](#) [Ready](#) [Review](#) [Verification](#) [Waiting on Reporter](#) [Workflow](#)

Details

Type: **+ Feature Request**  
Priority: **P3**  
  
Labels: **GM** **Go\_Monorepo** **arm64** **phab2t3\_label**  
Epic Link: **Hermetic Toolchain**  
Phab Type: **Story**  
Phab Link: **T3041491**

Description

We currently rely on bazel automatically detecting the c toolchain thats installed on the system using <https://blog.bazel.build/2016/03/31/autofconfiguration.html>.

This has a few issues:

People

Assignee: **Ken Micklas**  
[Assign to me](#)  
  
Reporter: **Robbert Van Ginkel**  
**[X] (Inactive)**  
  
Votes: **0** [Vote for this issue](#)  
Watchers: **14** [Stop watching this issue](#)

Dates

Created: 22/May/19 7:18 PM  
Updated: 28/Mar/22 11:13 AM

[Share](#) [Export](#)

2021 Jan: found this

https://andrewkelley.me/post/zig-cc-powerful-drop-in-replacement-gcc-clang.html

**Andrew Kelley**

- RSS Feed
- GitHub
- Twitter
- Email

programming language  
Sponsor @ \$5/month

## `zig cc`: a Powerful Drop-In Replacement for GCC/Clang

If you have heard of [Zig](#) before, you may know it as a promising new programming language which is ambitiously trying to overthrow C as the de-facto systems language. But did you know that it also can straight up compile C code?

This has been possible for a while, and you can see some [examples of this on the home page](#). What's new is that the `zig cc` sub-command is available, and it supports the same options as [Clang](#), which, in turn, supports the same options as [GCC](#).

Now, I'm sure you're feeling pretty skeptical right about now, so let me hook you real quick before I get into the juicy details.

### Clang and GCC cannot do this:

```
andy@ark ~/tmp> cat hello.c
```

... and started messing around.

2021 Jan: finding and reporting bugs to zig cc

- ▶ Reported bugs. Nothing happens.



2021 Jan: finding and reporting bugs to zig cc

- ▶ Reported bugs. Nothing happens.
- ▶ I start donating \$50/month to ZSF.

2021 Jan: finding and reporting bugs to zig cc

- ▶ Reported bugs. Nothing happens.
- ▶ I start donating \$50/month to ZSF.
- ▶ Nothing happens for a week.

## 2021 Feb: conversation in #zig:libera.chat

<motiejus> What is the protocol to "claim" the dev hours once donated?

<andrewrk> ZSF only accepts no-strings-attached donations

<andrewrk> did you get a different impression somewhere?

## 2021 Feb: conversation in #zig:libera.chat

<motiejus> What is the protocol to "claim" the dev hours once donated?

<andrewrk> ZSF only accepts no-strings-attached donations

<andrewrk> did you get a different impression somewhere?

Ooops.

## 2021 June: start of bazel-zig-cc

- ▶ Adam Bouhenguel (@ajbouh) creates a working bazel-zig-cc.

## 2021 June: start of bazel-zig-cc

- ▶ Adam Bouhenguel (@ajbouh) creates a working bazel-zig-cc.
- ▶ Jakub implements zig ar.

## 2021 June: start of bazel-zig-cc

- ▶ Adam Bouhenguel (@ajbouh) creates a working bazel-zig-cc.
- ▶ Jakub implements zig ar.
- ▶ I integrate zig ar and announce it:

# 2021 June: start of bazel-zig-cc

- ▶ Adam Bouhenguel (@ajbouh) creates a working bazel-zig-cc.
- ▶ Jakub implements zig ar.
- ▶ I integrate zig ar and announce it:
- ▶

[~andrewrk/ziglang](#)

archives

patches

new post

---

[ANN] bazel-zig-cc generally usable

[Motiejus Jakštys](#) <[motiejus@jakstys.lt](mailto:motiejus@jakstys.lt)>

▶ Details [7 months ago](#)

Hi folks,



I am announcing bazel-zig-cc[1]: a C++ toolchain for Bazel projects based on the venerable zig c++. It is generally usable: to register a default C++ toolchain for x86\_64 linux target, add a couple of lines[2] to the WORKSPACE.


# Usage





2021 July: start a “WIP diff”


2021 July: start a “WIP diff”


 Phabricator  941

 Differential > D6269911

 **zig c++ toolchain**

 Closed

 ACL: Differential users - ENG

 Authored by **motiejus** on Jul 1 2021, 17:43.

2021 End: little progress.

- ▶ The diff is up, with thousands of failing tests.

2021 End: little progress.

- ▶ The diff is up, with thousands of failing tests.
- ▶ Monorepo team doesn't show interest in picking it up.

## 2021 End: little progress.

- ▶ The diff is up, with thousands of failing tests.
- ▶ Monorepo team doesn't show interest in picking it up.
- ▶ I kept maintaining the diff at my "20% time".

## 2021: Recap

- ▶ Uber needs a hermetic C++ cross-compiler, but won't fund it.

## 2021: Recap

- ▶ Uber needs a hermetic C++ cross-compiler, but won't fund it.
- ▶ bazel-zig-cc kinda works, but nowhere near “real use”.

## 2021: Recap

- ▶ Uber needs a hermetic C++ cross-compiler, but won't fund it.
- ▶ bazel-zig-cc kinda works, but nowhere near “real use”.
- ▶ zig cc has bugs or missing features.



## 2021: Recap

- ▶ Uber needs a hermetic C++ cross-compiler, but won't fund it.
- ▶ bazel-zig-cc kinda works, but nowhere near “real use”.
- ▶ zig cc has bugs or missing features.
  - ▶ Donations don't “help”.

## 2021: Recap

- ▶ Uber needs a hermetic C++ cross-compiler, but won't fund it.
- ▶ bazel-zig-cc kinda works, but nowhere near “real use”.
- ▶ zig cc has bugs or missing features.
  - ▶ Donations don't “help”.
  - ▶ I can't realistically implement them.

## 2021: Recap

- ▶ Uber needs a hermetic C++ cross-compiler, but won't fund it.
- ▶ bazel-zig-cc kinda works, but nowhere near “real use”.
- ▶ zig cc has bugs or missing features.
  - ▶ Donations don't “help”.
  - ▶ I can't realistically implement them.
- ▶ “WIP diff” simmering and waiting for it's time.

2021 End: things start moving

I was asked/funded to evaluate arm64 to run our stack.

## 2021 End: things start moving

I was asked/funded to evaluate arm64 to run our stack.

Things started falling to place.

# Onboarding new technology in a megacorp

- ▶ Need a cross-compiler: `zig cc` may work.

# Onboarding new technology in a megacorp

- ▶ Need a cross-compiler: `zig cc` may work.
- ▶ Novel technology: need to de-risk bugs/missing features.

# Onboarding new technology in a megacorp

- ▶ Need a cross-compiler: `zig cc` may work.
- ▶ Novel technology: need to de-risk bugs/missing features.
  - ▶ Arranged a support contract.



# Onboarding new technology in a megacorp

- ▶ Need a cross-compiler: `zig cc` may work.
- ▶ Novel technology: need to de-risk bugs/missing features.
  - ▶ Arranged a support contract.
- ▶ Many internal discussions whether it's a good long-term fit.

# Onboarding new technology in a megacorp

- ▶ Need a cross-compiler: `zig cc` may work.
- ▶ Novel technology: need to de-risk bugs/missing features.
  - ▶ Arranged a support contract.
- ▶ Many internal discussions whether it's a good long-term fit.
  - ▶ This almost failed because “no large users, risky”.

# Onboarding new technology in a megacorp

- ▶ Need a cross-compiler: `zig cc` may work.
- ▶ Novel technology: need to de-risk bugs/missing features.
  - ▶ Arranged a support contract.
- ▶ Many internal discussions whether it's a good long-term fit.
  - ▶ This almost failed because “no large users, risky”.
  - ▶ MacOS cross-compiler won the argument.

# Support Contract

## Zig Software Foundation

### Transaction List by Date

January 1, 2021 - January 28, 2022

Date	Transaction Type	Name	Memo/Description	Split	Amount
2022-01-27	Expense	Jakub Konka	WT 210301-038438	Contractors	-8,000.00
2022-01-27	Expense	Wells Fargo	WIRE TRANS SVC	Bank Charges & Fees	-45.00
2022-01-24	Deposit	Github Sponsors	GitHub Sponsors	Donation Income	21,487.06
2022-01-21	Deposit	Uber	UBER USA,LLC	Support Contract	52,800.00

## 2022-02-02: the diff is landed



submitqueuebot added a comment.

Feb 2 2022, 20:44



Your merge request for target-branch main has been received by SubmitQueue. Please use the following URL: [/#/show/1928345](#) to check the status of your request



submitqueuebot added a comment.

Feb 2 2022, 20:52



Your merge request has been successfully merged.



submitqueuebot closed this revision.

Feb 2 2022, 20:52



submitqueue <udocker@

> added a commit:

rGOCODVJ98ecd9c761ee: zig c++ toolchain.

Feb 2 2022, 20:53

## Next steps

- ▶ Enable by default for all builds.

## Next steps

- ▶ Enable by default for all builds.
- ▶ Make it work/enable for MacOS targets.

## How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.



## How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.
  - ▶ megabytes: nice to have, engineers like small things.

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.
  - ▶ megabytes: nice to have, engineers like small things.
  - ▶ demonstrate-able cross-compiler for osx: on the roadmap.

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.
  - ▶ megabytes: nice to have, engineers like small things.
  - ▶ demonstrate-able cross-compiler for osx: on the roadmap.
- ▶ Teams and my director did not trust me as much as they did.

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.
  - ▶ megabytes: nice to have, engineers like small things.
  - ▶ demonstrate-able cross-compiler for osx: on the roadmap.
- ▶ Teams and my director did not trust me as much as they did.
- ▶ We didn't have the budget *at that time*.



# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.
  - ▶ megabytes: nice to have, engineers like small things.
  - ▶ demonstrate-able cross-compiler for osx: on the roadmap.
- ▶ Teams and my director did not trust me as much as they did.
- ▶ We didn't have the budget *at that time*.
- ▶ Andrew wouldn't have signed the contract.

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.
  - ▶ megabytes: nice to have, engineers like small things.
  - ▶ demonstrate-able cross-compiler for osx: on the roadmap.
- ▶ Teams and my director did not trust me as much as they did.
- ▶ We didn't have the budget *at that time*.
- ▶ Andrew wouldn't have signed the contract.
- ▶ If preparation were not here.

# How could this have failed?

For **any** of the reasons below:

- ▶ I wouldn't have been annoyed with status quo.
- ▶ I wouldn't have found Andrew's post about "universal compiler".
  - ▶ Make sure you are visible. Good job, Andrew and Loris.
- ▶ The product was not much better than the competition:
  - ▶ specifying the glibc version: very important.
  - ▶ megabytes: nice to have, engineers like small things.
  - ▶ demonstrate-able cross-compiler for osx: on the roadmap.
- ▶ Teams and my director did not trust me as much as they did.
- ▶ We didn't have the budget *at that time*.
- ▶ Andrew wouldn't have signed the contract.
- ▶ If preparation were not here.
  - ▶ When we needed a cross-compiler, time was tight.

# TLDR

- ▶ Preparation (including lots of grinding).

# TLDR

- ▶ Preparation (including lots of grinding).
- ▶ Timing.

# TLDR

- ▶ Preparation (including lots of grinding).
- ▶ Timing.
- ▶ Lots of luck.

# TLDR

- ▶ Preparation (including lots of grinding).
- ▶ Timing.
- ▶ Lots of luck.

# TLDR

- ▶ Preparation (including lots of grinding).
- ▶ Timing.
- ▶ Lots of luck.



# Questions & Appendix

{