

# **Build It! With or Without Tools**

Matthew Sanabria



# Matthew Sanabria

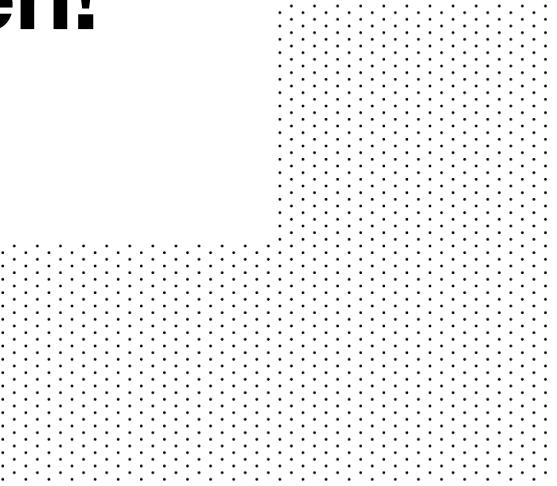
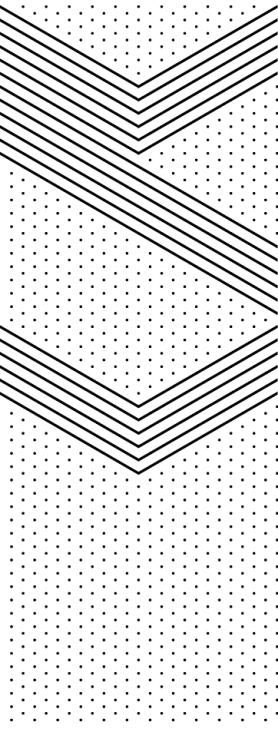
Staff Site Reliability Engineer  
Cockroach Labs

[matthewsanabria.dev](http://matthewsanabria.dev)  
@sudomateo





**Build what, exactly?**



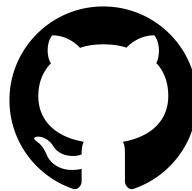
**How about ~~production~~  
~~applications~~ a kitchen!**



# The Tools (Hardware)



# The Tools (Software)



# The Tools (Software)



# Weighing Options

## What can I do in this situation?

### Do Nothing

Live with the disdain for the current kitchen.

Be constantly reminded that the kitchen needs renovating.

### Hire a Contractor

Costs a ton of money that I don't have because I just bought a house.

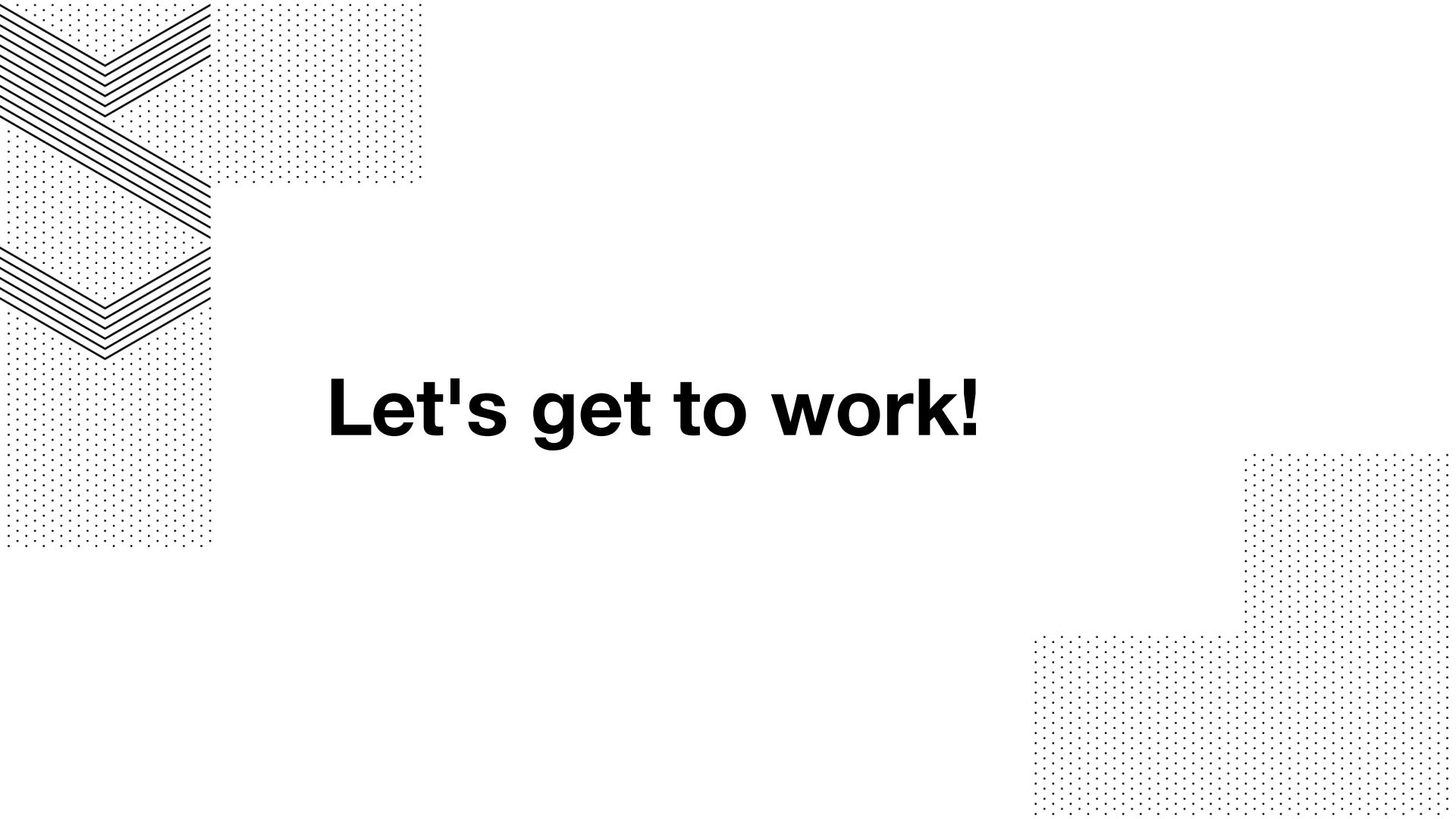
Won't learn any new skills or better understand how my house is built.

### Build it Myself

Spend a lot of time working after-hours. I still have a day job after all.

Save a ton of money.

Learn new skills and better understand how my house is built.

The background features abstract geometric patterns. On the left, there's a vertical column of chevron-like shapes pointing downwards, composed of thin black lines. To the right of this, a large area is filled with a uniform grid of small black dots. In the bottom right corner, there's another vertical column of chevron shapes pointing upwards, also made of thin black lines.

**Let's get to work!**

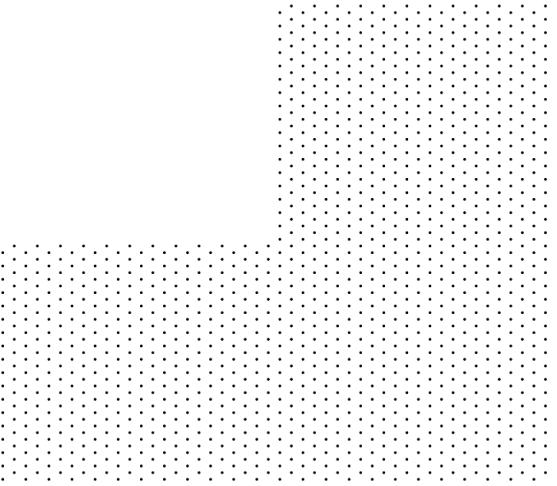
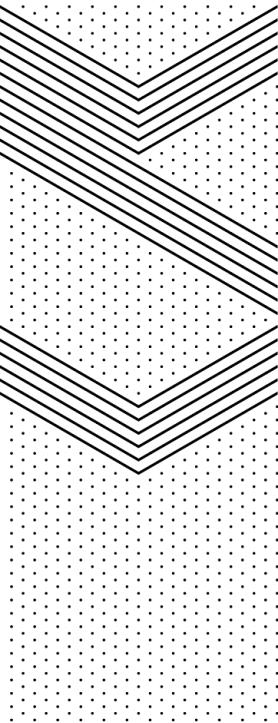






The background features abstract geometric patterns. On the left side, there is a vertical column of chevron-like shapes pointing downwards, composed of thin black lines. To the right of this, a large area is filled with a uniform dot pattern. In the bottom right corner, there is another vertical column of chevron-like shapes pointing upwards, also composed of thin black lines.

**There's always  
something.**



# **Scenario #1 - Right tools. Wrong knowledge.**



**What specs??  
How work??**

NATIONAL ELECTRICAL CODE®

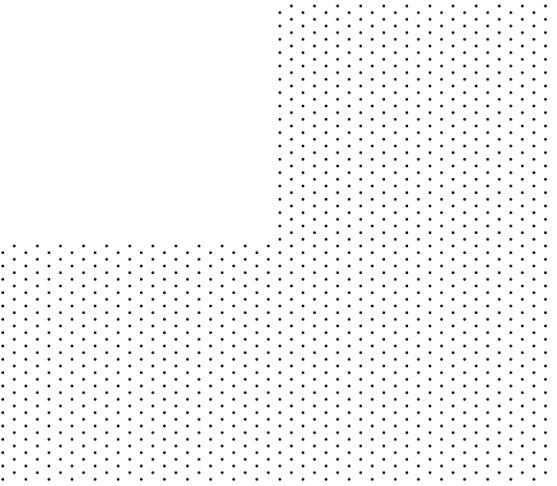
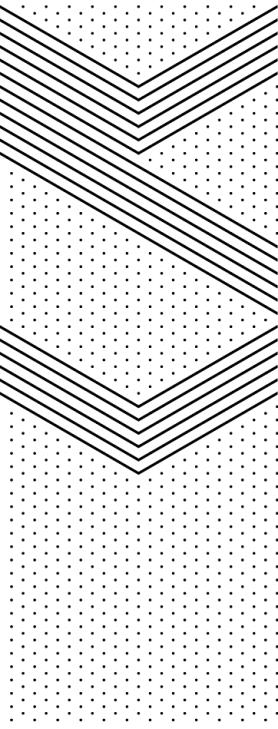


NFPA 70

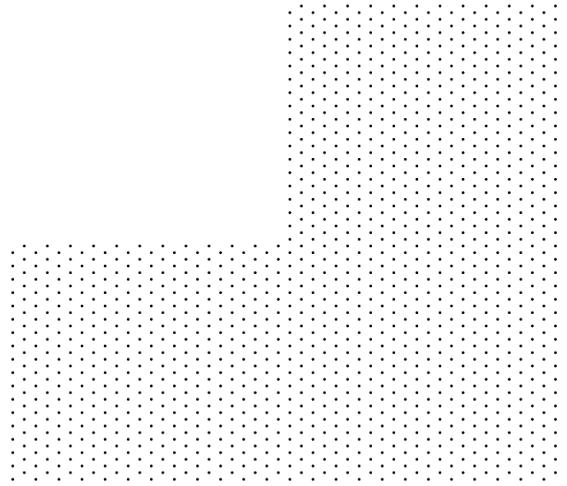
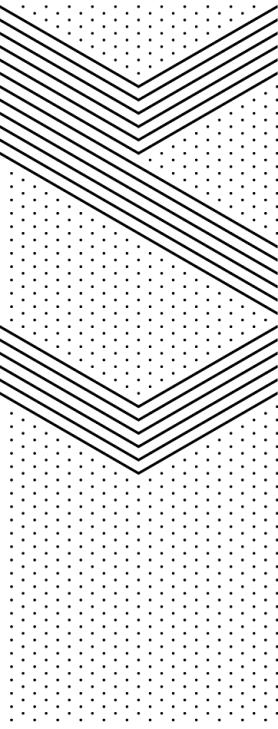
# Go Documentation

## Go's version of the NEC.

- <https://pkg.go.dev/> - Documentation for Go packages and standard library.
  - Look for this button on GitHub:  reference
- <https://go.dev/ref/spec> - Go language specification.
- go doc - CLI command to generate documentation for source code.
  - Only helpful if you write documentation comments in the first place.



**Documentation adds  
new tools to the  
toolbox.**



## **Scenario #2 - Wrong tools. Write tests.**



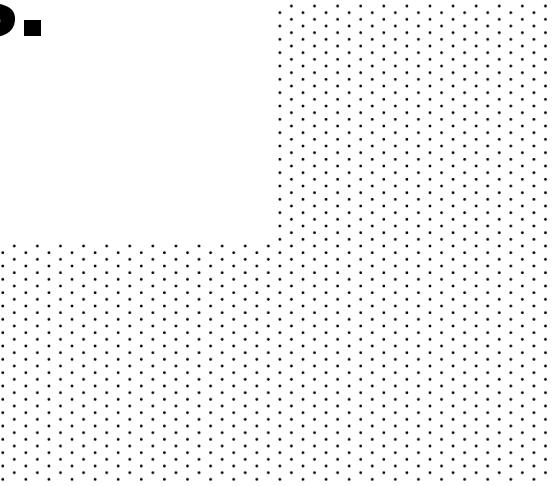
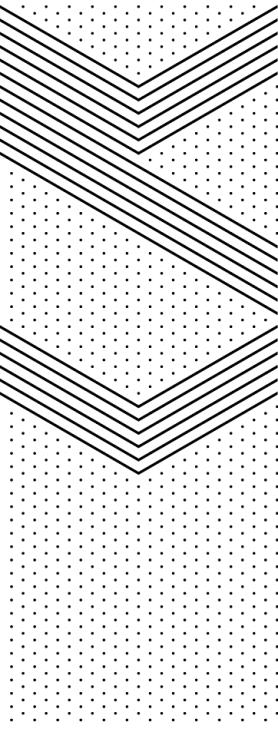




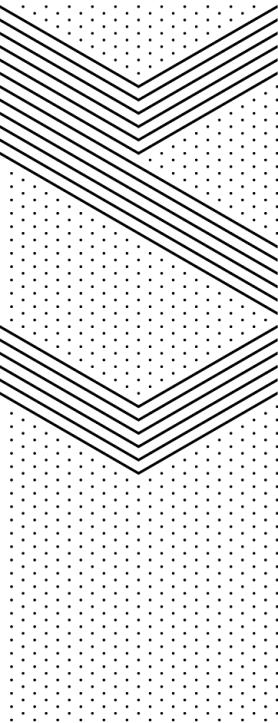
# Troubleshooting in Go

**Finding, fixing, and testing incorrect behavior.**

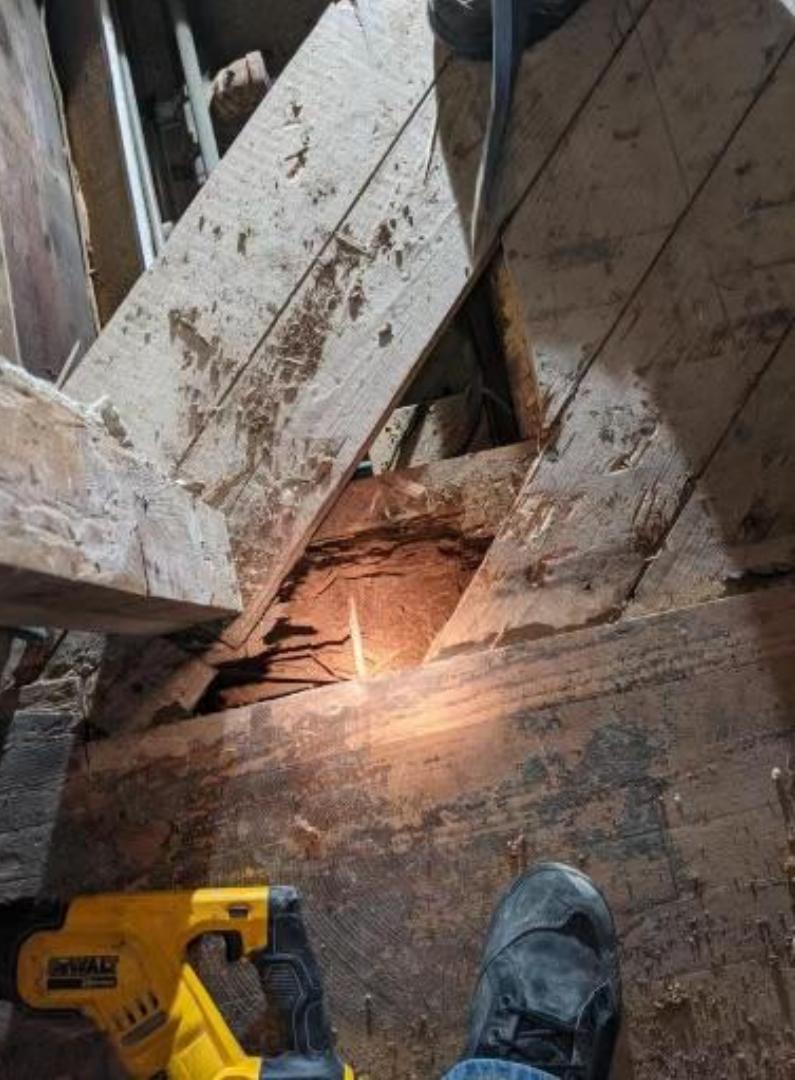
- go test - CLI command to test Go programs.
  - <https://pkg.go.dev/testing> - Package for testing, benchmarking, and fuzzing.
- go tool pprof - CLI command to interact with profiling data.
  - <https://pkg.go.dev/net/http/pprof> - Package for generating profiling data.
- <https://github.com/go-delve/delve> - A debugger for Go.



**Verify tools work  
correctly with tests.**



# **Scenario #3 - Wrong tools. Wrong knowledge. No time.**

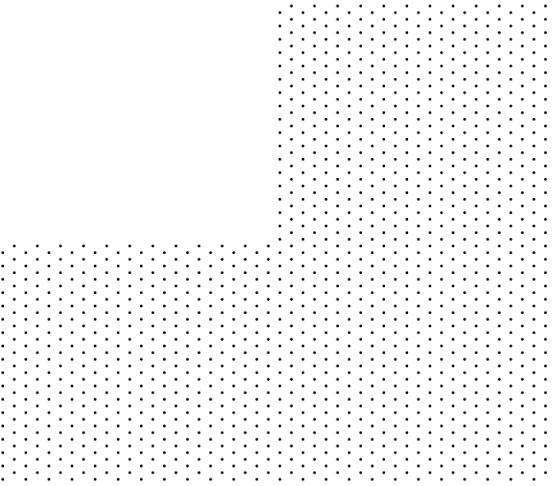
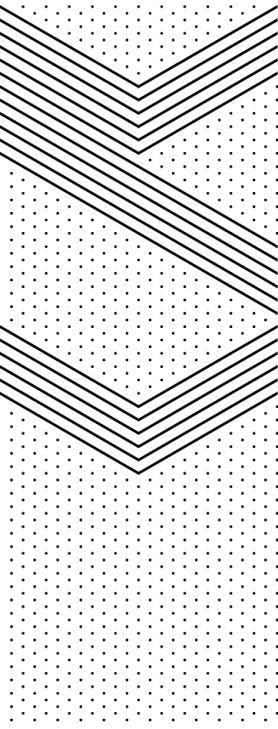




# Avoiding Go

**It's okay to say no to loved ones.**

- Go isn't always a good fit.
  - Frontend & Mobile applications
  - Existing ecosystem requirements
  - Lack of expertise with advanced features
- Seek help from more experienced engineers but **observe their work.**



**Know when to seek  
an expert. Observe  
their work.**



Is the kitchen done?









# **Build It!**

## **Closing thoughts.**

- Building helps you learn more about your system and expand the tools in your toolbox.
- When tools are limited allow your creativity to guide you.
  - It's amazing what you can accomplish with old, boring tools.
- Understand what tools experts use in your industry.
- The greatest tool in any toolbox... is you.



**Thank you!**

**matthewsanabria.dev**  
**@sudomateo**