# Developer Screening Process - Phase 1

## Overview

* Write a project in Go (golang)
  + A stub of main.go has been provided - in this file is some addition background information on the hypothetical scenario as well as the example data you’ll be working with.
* Track changes with git
  + At least once at the start and once when you’re finished
    - Bonus points if you track your changes as you develop your solution - though don’t force it if it doesn’t feel natural.
  + Bonus points if you upload your program to github
* When finished
  + Submit the output of the project to [jacob.schmit@theice.com](mailto:jacob.schmit@theice.com)
  + Share either the github link of the project or provide additional output:
    - The output of git status
    - The output of git log

## Expected Output

The output you program should product is (with actual values in place of the VALUEs):

Sample Count: VALUE

Peak Max Bytes Observed: VALUE

Was observed at this ms epoch: VALUE

Peak microburst in bps: VALUE

Peak microburst Mbps: VALUE

Mean Bytes Observed: VALUE

Total Bytes Observed: VALUE

If you do not upload the program to github please also submit:

The output of git status

The output of git log

## Requirements/Constraints

1. Install and setup a golang development environment. Go strongly suggests a certain project structure for the ecosystem to work best.
2. There are no restrictions on your use of the internet - please try to not copy and paste a solution together though!
3. There are no restrictions on using an IDE vs text editor
4. If you decide to upload your project to github please use the project name ICE5KEMDKF
   1. You can delete this once you are done with the screening process
5. Please try to always run gofmt and golint - you don’t have to fix every warning/notice but nothing major should be produced

## FAQ

Q: Why Go?

A: Go is a relatively new language and as such is a good guage on someones ability to shift gears and use something they might not have used before. Go is c-like but has a substainly easier tool chain bringing it closer to working with something like Python.

Q: Why track with git?

A: Version control is an import tool - demonstrating a basic familiarity will show that joining projects that are already under version control should be relatively easy.

Q: Why suggest uploading to github?

A: This will demonstrate that you not only have the basics of git down but you understand the push/pull methodology of updating a remote repository. Depending on how you set up the github account it could also demonstrate familiarity with ssh key authentication.