

# Project documentation

**Description:** Zcodesystem.com [1] Hockey and NHL scrapper with Excel [2] and Markdown [3] generation and daily deliver conditional results using Telegram [4].

The project has been requested on freelancer.com [5] platform and resulted on automation program written in GO language [6].

*I Need a script to take automatically all the data from the table here --  
> <https://zcodesystem.com/scorespredictor/>(Hockey and NHL)*

*Put everything in a excel file (Table 1) that I receive in some ways like email  
or other ways, you maybe suggest me the best way.*

*Select the lines from Table 1 where a team should win with 4 or more goal  
so for example 0-4 1-5 etc. AND with "confidence" >69 %*

*The result must go on a Table 2 or another Excel File.*

*All the lines of Table 2 should be AUTOMATICALLY send on a Telegram  
Channel with a Telegram Bot that auto publish every line of the Table 2,  
every day at the same hour.*

*So usually Table 2 on excel should have 4-5 lines because every line must  
respect the conditions ( Confidence > 69% and goal difference >3 )*

*Nicola Conticello*

Developed by: Filipe Ferreira

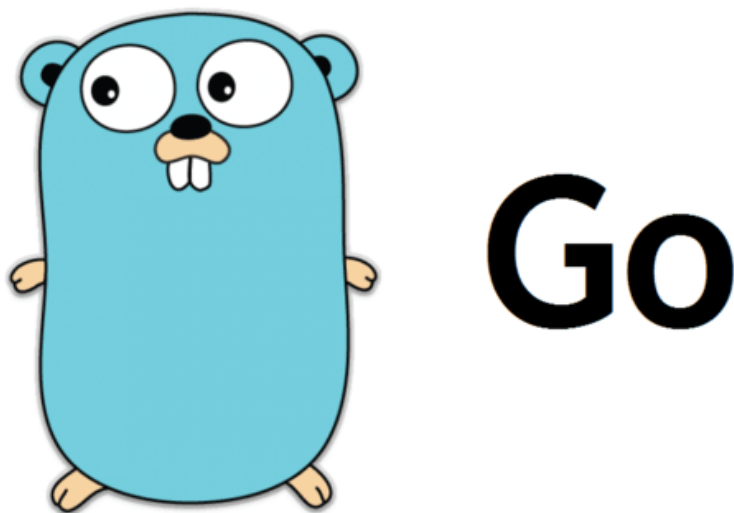
Monday, 4 March 2019

# 1. Code

The code was written in GO language [6] and is split into small files representing each functionality (Table 1 - Code structure).

*Table 1 - Code structure*

Scorespredictor.go	Entrypoint an main logic
Config.go	Load the configuration from .env.yaml [7] file
Fetch.go	Call the website endpoint [1]
Excel.go	Generate the excel [2]
Db.go	Create and seed small SQLITE [8] database
Telegram.go	Init the bot [4], generate and send markdown message [3] and file
Log.go and Explorer.go	Helpers to application text output and delete a file



*Figure 1- Go*

## 2. Usage guide

A GO [6] application can be run issuing the command “go run” in the script directory or compiled into an executable issuing the “go build” command.

In order to issue the commands, the go runtime must be installed [9] and a command line instance must be opened in the script directory.

Also note that the script have some web dependences that must first be fetch. Issuing the “go build” command with “-i” flag most of time is enough, otherwise run “go install”.

Was created an env file written in YAML [7] to make some settings easy to change without needing to dig into the code (Table 2- .env.yaml).

Table 2- .env.yaml

SCORESPREDICTOR_URL	The endpoint [1] used to fetch data
SCORESPREDICTOR_SPORTS	A comma “,” separated string with the sport pages that will be fetched
TELEGRAM_BOT_TOKEN	The token necessary to use the telegram bot [4].
TELEGRAM_BOT_CHANNEL	Chanel used to send the markdown [3] report
TELEGRAM_BOT_CHANNEL2	Chanel used to send the excel [2] file
FILTER	A comma “,” separated string contains the queries used to filter the results. The order of the queries is the same as used in SCORESPREDICTOR_SPORTS.

**Note:** For private channels we must first get the channel ID. In order to do that, login under your account at web version of Telegram [10] and find your channel. See to your url, it should be like [https://web.telegram.org/#/im?p=c1055587116\\_11052224402541910257](https://web.telegram.org/#/im?p=c1055587116_11052224402541910257)

Grab "1055587116" from it, and add "-100" as a prefix.

Your channel id will be "-1001055587116".

### 3. Google Cloud Platform

In order to run the script every day we created a new project in GCP [11] and inside the project we created a Compute Engine [12] minimal always free [13] vm instance - Figure 1- F1 micro instance with centos [14] operative system.

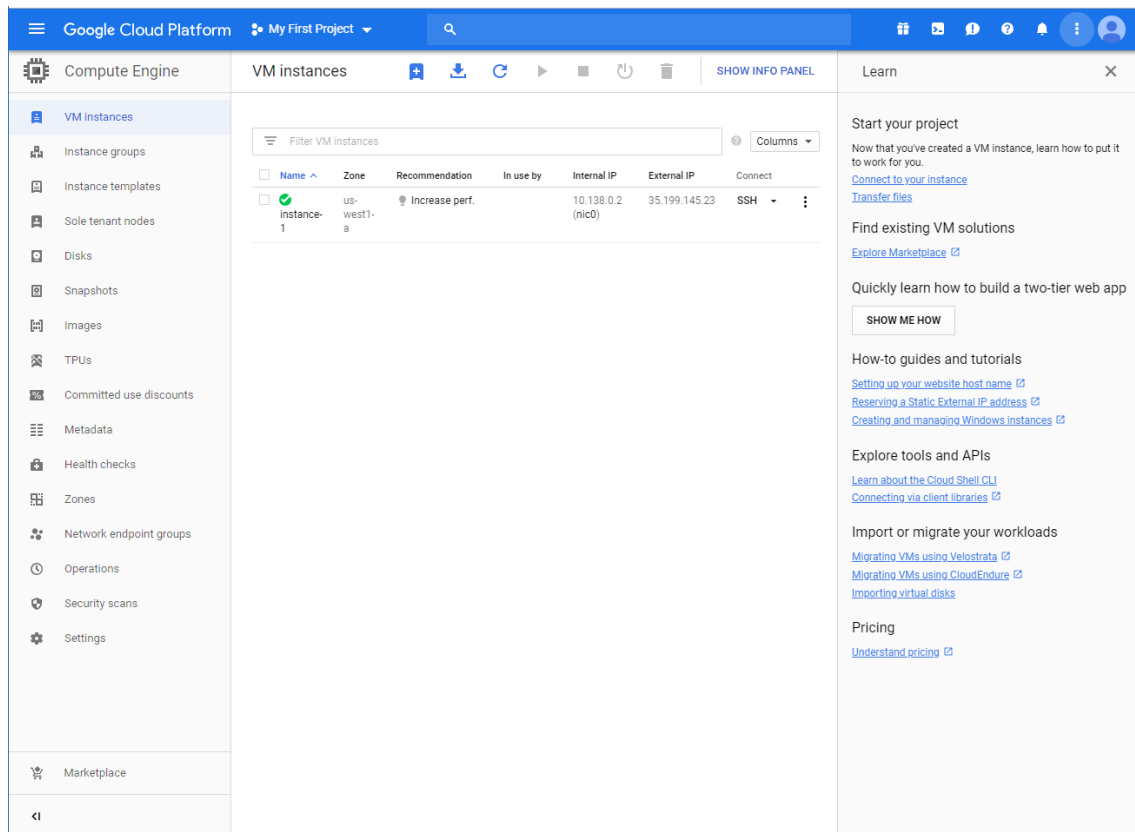


Figure 2- F1 micro instance

After we have created the vm, we installed GO [6] and setup the cron to run every day at 10am Italy time zone. Then, the application has been placed in /home/nicola/source/scorespredictor/ folder.

The commands issued to configure the SO and the application can be seen in Code 1 - Centos configuration.

```
#!/usr/bin/env bash

sudo yum update && sudo yum install wget nano
wget https://dl.google.com/go/go1.12.linux-amd64.tar.gz
sudo tar -C /usr/local -xzf go*
sudo timedatectl set-timezone Europe/Rome

#-----
# /etc/profile
#-----
sudo sh -c ' cat <<EOT >> /etc/profile
export SCORESPREDICTOR_HOME=/home/nicola/source/scorespredictor/
EOT
' && source /etc/profile

#-----
# /etc/cron.d/scorespredictor
#-----
sudo sh -c ' cat <<EOT > /etc/cron.d/scorespresdictor
SHELL=/bin/bash
MAILTO=nicola
CRON_TZ=Europe/Rome

0 10 * * * nicola source /etc/profile && cd
/home/nicola/source/scorespredictor/ && go build -i -o compiled &&
./compiled >/dev/null
EOT
' && sudo systemctl restart crond.service
```

Code 1 - Centos configuration

In order to put the application into the remote machine and run the commands we generated an ssh key pair [15] and authorized it putting the public key in GCP Figure 2- GCP SSH metadata[11].

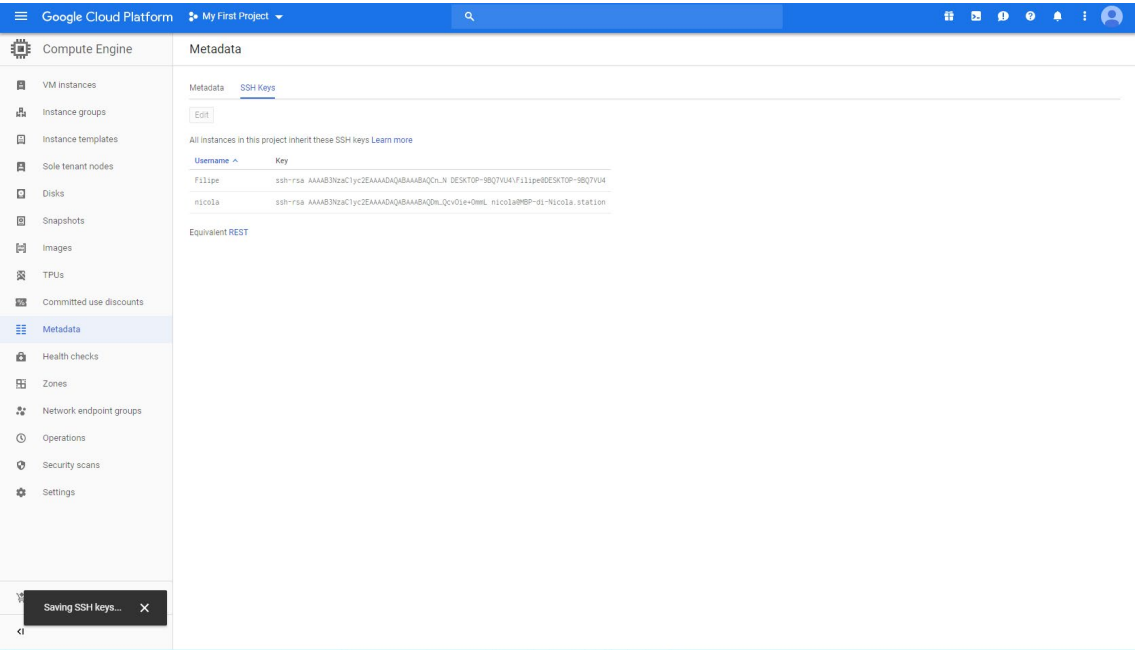


Figure 3- GCP SSH metadata

We then connected using terminal command `ssh username@instancepublicip -i ~/.ssh/id_rsa` and used GIT [16] clone it into the server.

**Note:** There are many graphical clients that can be used for file transfers. We recommended Cyberduck [17] for osx.

# References

- [1] "Z-Code System." [Online]. Available: <https://zcodesystem.com/>. [Accessed: 11-Mar-2019].
- [2] "Microsoft Excel." [Online]. Available: <https://products.office.com/pt-pt/excel>. [Accessed: 11-Mar-2019].
- [3] "Telegram Bot API Markdown." [Online]. Available: <https://core.telegram.org/bots/api#markdown-style>. [Accessed: 11-Mar-2019].
- [4] "Telegram Messenger." [Online]. Available: <https://telegram.org/>. [Accessed: 10-Mar-2019].
- [5] "Freelancer.com." [Online]. Available: <https://www.freelancer.com/dashboard>. [Accessed: 11-Mar-2019].
- [6] "The Go Programming Language." [Online]. Available: <https://golang.org/>. [Accessed: 11-Mar-2019].
- [7] "YAML™ Version 1.2." [Online]. Available: <https://yaml.org/spec/1.2/spec.html>. [Accessed: 11-Mar-2019].
- [8] "SQLite." [Online]. Available: <https://www.sqlite.org/index.html>. [Accessed: 11-Mar-2019].
- [9] "Getting Started - The Go Programming Language." [Online]. Available: <https://golang.org/doc/install>. [Accessed: 11-Mar-2019].
- [10] "Telegram Web." [Online]. Available: <https://web.telegram.org/#/im>. [Accessed: 11-Mar-2019].
- [11] "Google Cloud Platform." [Online]. Available: <https://cloud.google.com/>. [Accessed: 11-Mar-2019].
- [12] "Compute Engine." [Online]. Available: <https://cloud.google.com/compute/>. [Accessed: 11-Mar-2019].
- [13] "GCP free tier." [Online]. Available: <https://cloud.google.com/free/>. [Accessed: 11-Mar-2019].
- [14] "CentOS Project." [Online]. Available: <https://www.centos.org/>. [Accessed: 11-Mar-2019].
- [15] "How can I generate SSH keys on Mac OS X?." [Online]. Available: <https://secure.vexxhost.com/billing/knowledgebase/171/How-can-I-generate-SSH-keys-on-Mac-OS-X.html>. [Accessed: 11-Mar-2019].
- [16] "Git." [Online]. Available: <https://git-scm.com/>. [Accessed: 11-Mar-2019].
- [17] "Cyberduck - SFTP/FTP Client for Mac." [Online]. Available: <https://www.ssh.com/ssh/cyberduck/>. [Accessed: 11-Mar-2019].