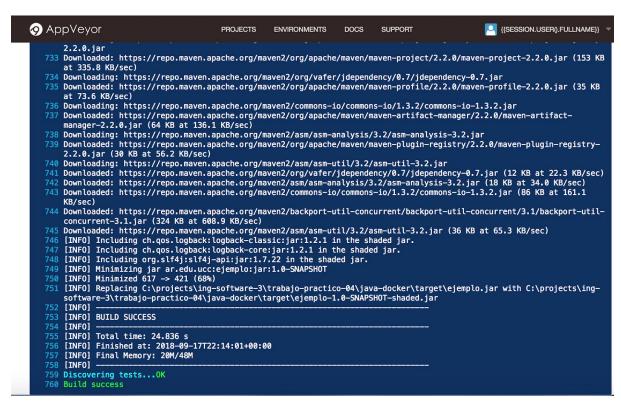
## 2- Configurando AppVeyor

- Crear una cuenta y evaluar que pasa en cada una de las herramientas. Usar el login con github para entrar a la primera.
- Configuramos el proyecto de Maven





## Artifacts

PATH TO ARTIFACT	DEPLOYMENT NAME	TYPE	
trabajo-practico-04\java-docker\target\ejemp	SalidaBuild	Auto	r

Add artifact

Save

```
AppVeyor
                                                                                              PROJECTS
                                                                                                                       ENVIRONMENTS
                                                                                                                                                           DOCS
                                                                                                                                                                             SUPPORT
                                                                                                                                                                                                                                 {SESSION.USER().FULLNAME}}
               tree-2.0.jar (55 KB at 120.8 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/commons-io/commons-io/1.3.2/commons-io-1.3.2.jar
               Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven-project/2.2.0/maven-project-2.2.0.jar (153 KB
                at 325.1 KB/sec)

    Tab/Sec/
    Townloading: https://repo.maven.apache.org/maven2/asm/asm-analysis/3.2/asm-analysis-3.2.jar
    Downloaded: https://repo.maven.apache.org/maven2/backport-util-concurrent/backport-util-concurrent/3.1/backport-util-concurrent-3.1.jar (324 KB at 690.7 KB/sec)
    Downloading: https://repo.maven.apache.org/maven2/asm/asm-util/3.2/asm-util-3.2.jar
    Downloaded: https://repo.maven.apache.org/maven2/org/vafer/jdependency/0.7/jdependency-0.7.jar (12 KB at 23.8 KB/sec)
    Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-registry/2.2.0/maven-plugin-registry-2.9 iar (30 KB at 50.8 KB/sec)

       2.2.0.jar (30 KB at 59.8 KB/sec)

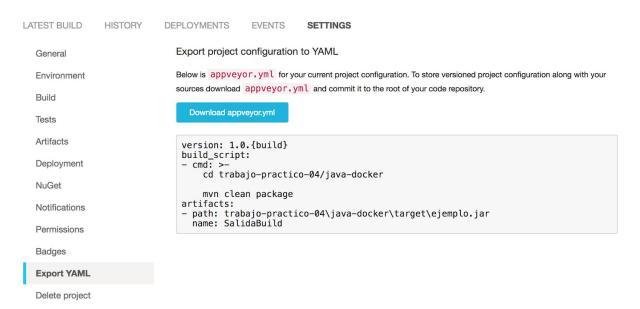
743 Downloaded: https://repo.maven.apache.org/maven2/commons-io/commons-io/1.3.2/commons-io-1.3.2.jar (86 KB at 171.4 KB/sec)

744 Downloaded: https://repo.maven.apache.org/maven2/commons-io/commons-io/1.3.2/commons-io-1.3.2.jar (86 KB at 171.4 KB/sec)
      KB/sec)
745 Downloaded: https://repo.maven.apache.org/maven2/asm/asm-analysis/3.2/asm-analysis-3.2.jar (18 KB at 35.1 KB/sec)
746 [INFO] Including ch.qos.logback:logback-classic:jar:1.2.1 in the shaded jar.
747 [INFO] Including org.slf4j:slf4j-api:jar:1.7.22 in the shaded jar.
748 [INFO] Including org.slf4j:slf4j-api:jar:1.7.22 in the shaded jar.
749 [INFO] Minimizing jar ar.edu.ucc:ejemplo:jar:1.0-SNAPSHOT
750 [INFO] Minimized 617 -> 421 (68%)
751 [INFO] Replacing C:\projects\ing-software-3\trabajo-practico-04\java-docker\target\ejemplo-jar with C:\projects\ing-
752 [INFO]
                 [INFO] Total time: 19.113 s
[INFO] Finished at: 2018-09-17T22:26:12+00:00
                                Final Memory: 19M/47M
                 [INFO]
               Collecting artifacts...
Found artifact 'trabajo-practico-04\java-docker\target\ejemplo.jar' matching 'trabajo-practico-04\java-docker\target\ejemplo.jar' path
                Discovering tests...0
                [1/1] trabajo-practico-04\java-docker\target\ejemplo.jar (559,637 bytes)...100%
```

Acá podemos generar los builds y publicarlos

# Para ejecutarlo: java -jar ejemplo.jar

[MacBook-Pro-de-Laura:Downloads laurapognante\$ java -jar ejemplo.jar 19:31:35.722 [main] INFO ar.edu.ucc.App - Hola Mundo!



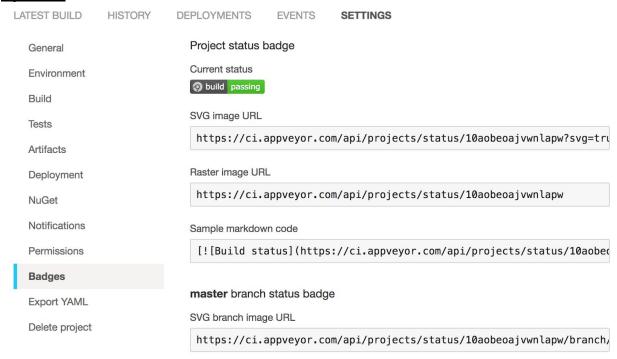
#### PORQUE UN YAML???

#### Secreto:

 En todo momento como lo tengo en git bajo control de configuración, se como aplique la configuración y como la construi. Si necesito volver 6 meses atras y saber que use para construirlo, puedo volver y hacerlo.

- Si viene alguien nuevo y quiere usarlo, con esto hereda del mio la configuración.
   Seteando en mi propia tool, y no tiene que hacer nada más que correrlo, haciendo solamente el build.
- Cuando lo pongo en mi repo ya sabe cómo construirse

## Optional:



## si pongo en mi repo:

[![Build

status](https://ci.appveyor.com/api/projects/status/10aobeoajvwnl
apw?svg=true)](https://ci.appveyor.com/project/lalipognante/ing-s
oftware-3)

 Con un BADGES: Le voy a hacer saber a la gente que mi build y mi codigo complica.

Cuando entro al README voy a saber muchas cosas de mi repo. Por ejemplo que el build está corriendo.