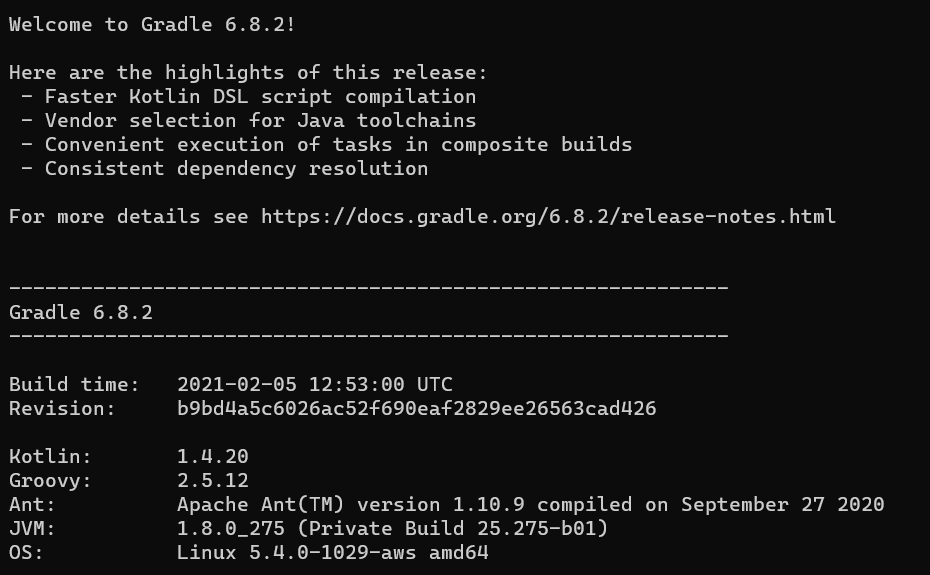
**TALLER 1**

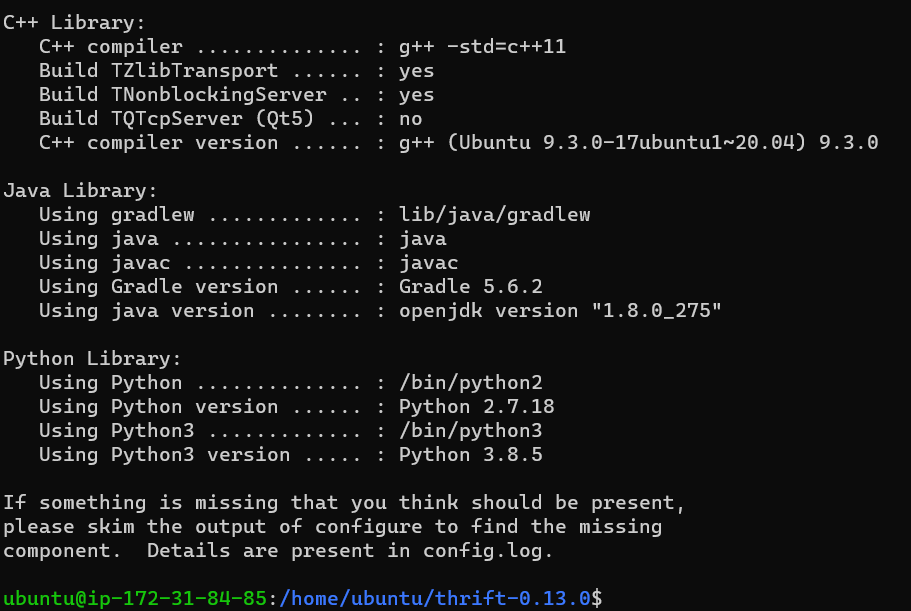
**BIG DATA**

**Víctor Samuel Pérez Díaz**

1. **10.**

****

**16.**

****

1. **1.**Thrift soporta comentarios tipo C style (/\* comment \*/) y C simple (// comment), así como comentarios con # (# comment).  
     
   **2.**

* **bool Booleano de un bit**
* **i8 (byte) Entero de 8 bits con signo**
* **i16 Entero de 16 bits con signo**
* **i32 Entero de 32 bits con signo**
* **i64 Entero de 64 bits con signo**
* **double Valor de punto flotante de 64 bits**
* **string Cadena de caracteres**
* **binary Blob (Objetos Binarios Grandes) (arreglo de bytes)**
* **map<t1,t2> Mapa de un tipo a otro**
* **list<t1> Lista ordenada de un tipo**
* **set<t1> Conjunto de elementos únicos de un tipo**

**3.**

Se pueden incluir otros archivos Thrift para incluir estructuras comunes o definiciones de servicios. Estos se pueden encontrar en el path actual o buscando en otro path especificado con la flag de compilación -I.

Se puede acceder a estos archivos usando el nombre del .thrift como prefijo.

Ejemplo: shared.thrift

**4.**

**namespace java tutorial**

**namespace py tutorial**

**5.**

**Lenguajes soportados:**

ActionScript  
C (glib)

C++  
Common LISP

Dlang

Dart

Delphi  
.NET

Standard

Erlang

Go

Haskell

Haxe

Java (SE)

Java (ME)

Javascript

Lua

Node.js

Node.ts

OCaml

Perl

PHP

Python

Ruby

Rust

Smalltalk

Swift

**6.** const list<int> num= {1,2,3,4,5}

**7.**

enum french\_suite {

DIAMOND = 1,

CLUB = 2,

HEART = 3,

SPADE = 4

}

**8.**

struct Programa {

1: i64 ID,

2: string nombre,

3: string apellido,

4: optional string programa,

}

**9.**

service BigData {

void ping(),

bool add(1:string nombre, 2:string apellido),

float compute(1:float fl1, 2:float fl2) throws (1:InvalidOperation ups),

oneway void receive(string str)

}