

## Guía de despliegue de TP



*"Encontré un leak el viernes antes de la entrega" - Pintura al óleo.*

### Objetivo

---

Esta guía está lejos de estar redactada para que ustedes adquieran nuevos conocimientos de la materia. Su propósito es más sencillo, es más bien una lista de tareas o checklist de todas las cosas que deberán hacer el día de la entrega de su TP, junto con algunas pautas y recomendaciones para poder agilizar su despliegue lo más posible. Todo lo que sea obligatorio y opcional/recomendado estará marcado como tal.

### Paso 0: Preentrega - Recomendado

---

No toda tarea que se haga en la entrega tiene que hacerse. Hay muchas cosas que pueden adelantarse, y eso es algo que nos va a servir a todos para agilizar los tiempos de evaluación y hacer que la menor cantidad de gente tenga que esperar para entregar.

## Generar las credenciales

Para empezar, recomendamos ampliamente haber generado un **Personal Access Token de GitHub** [🔗](#) que **expire al día siguiente** <sup>[1]</sup> y tenerlo a mano antes de entrar al laboratorio para poder clonar el repositorio del TP. Sí, tarda 5 minutos en crearse, pero si multiplicamos ese tiempo por los 80 grupos por entrega se va a notar de cuánto tiempo ahorrado para todos estamos hablando.

## Contar con los config de las pruebas

En segundo lugar, los archivos de configuración: cada prueba especifica los parámetros con los que debe contar cada módulo, entonces podrían guardar en el repositorio un archivo distinto por cada prueba y **pasar el que corresponde por línea de comandos**. De esta forma, para pasar a la siguiente simplemente bajamos y le pasamos un archivo de configuración distinto al mismo ejecutable.

## Practicar

Otro aspecto importante es interiorizarse sobre el método de despliegue que van a querer en su TP.

### IMPORTANTE

Las entregas son en las mismas VMs server **SIN INTERFAZ GRÁFICA** que les proporcionamos a través de los **links de descarga**, por lo que van a tener que aprender un mínimo de uso de la consola.

Si tuviéramos que resumir en tres cosas, lo que deberían hacer para poder llegar a una entrega con la seguridad de que van a aprobar sería:

1. Probar el TP
2. Practicar el despliegue
3. Probar el TP y practicar el despliegue

Para esto pueden ir a los laboratorios de Medrano (que están abiertos de lunes a viernes de 10 a 21hs y sábados de 10 a 18hs) o, de no ser posible, pueden utilizar las VMs server desde sus casas. Para ambos casos, más adelante en la guía les mostraremos algunas herramientas que pueden serles de utilidad 😊

Prueben el TP de la misma manera que nosotros lo vamos a estar haciendo, midanse sus tiempos y practiquen con la consola, revisen qué es lo que más les cuesta o tiempo les consume y fíjense una manera de reducirlos al máximo.

### TIP

Notarán que las tareas a realizar son siempre las mismas, por lo que pueden automatizar el procedimiento armando un **script de Bash** [🔗](#).

Ante cualquier duda, estamos para ayudarlos 😊

## Paso 0.1: Dormir - Obligatorio

No tiene más presentación. Sabemos y entendemos que la entrega de algo en lo que llevan poniendo tanto esfuerzo puede causar nervios y estrés, pero el venir lo más frescos y despejados posibles a una situación es un ingrediente importante del éxito.

## Paso 1: Anunciarse - Obligatorio

---

¡Llegó el gran día! Es hora de comenzar el despliegue y maravillar a los ayudantes con estos tres meses de trabajo... Casi.

Antes de poder hacer nada van a tener que anunciar que llegaron y están listos para entregar. Estamos limitados por la cantidad de computadoras del laboratorio, por lo que para los turnos más tardíos deberán esperar más.

La metodología para anunciarse se las vamos a compartir unos días antes de la entrega

## Paso 2: Despliegue - Obligatorio

---

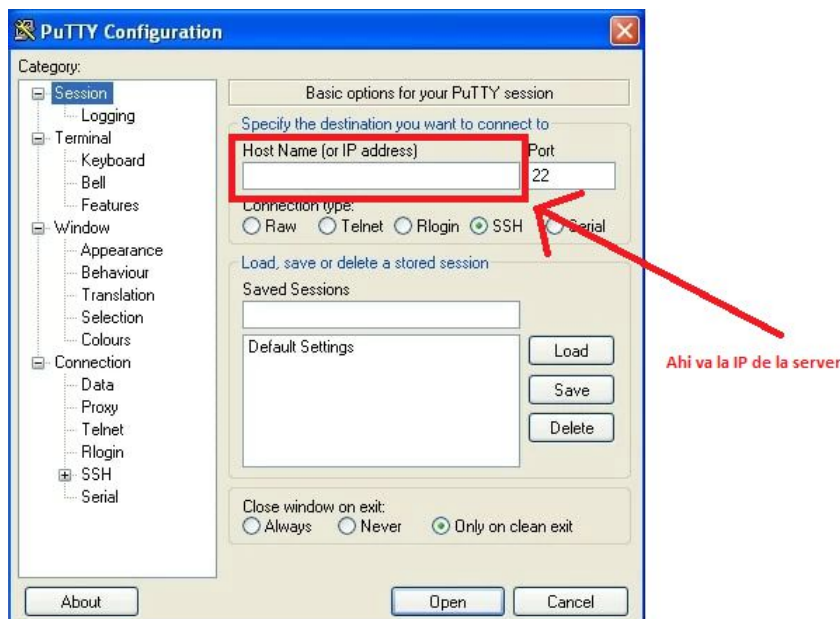
Ahora sí, llegó el momento. Cuando entren al laboratorio los vamos a dirigir a la fila que les toca y van a encontrarse con (aproximadamente) tres computadoras con una VM Server abierta.

Es importante que sepan que el despliegue **no puede demorar más de 20 minutos**. Puede haber margen de tolerancia, pero es un tiempo de descuento que irá en función de cuantos otros grupos estén esperando para entregar.

### PuTTY

Es altamente probable que necesiten usar el portapapeles desde Windows para ingresar el Personal Access Token y clonar el repo. Incluso, más adelante les pediremos iniciar varias consolas sobre cada VM Server. El método más sencillo para lograr ambas cosas sin una interfaz gráfica es conectarse por SSH.

Desde Windows (que es el entorno sobre el que están instalado VirtualBox en el laboratorio), la forma más sencilla de hacerlo es utilizando **PuTTY** [↗](#):



## Pasos a seguir

A partir de acá, lo que queda es usar la consola para clonar, compilar, configurar y levantar sus módulos. Algunas cosas a recordar en esta parte:

1. Clonar el repo
2. Instalar las bibliotecas necesarias (incluyendo las **commons** <sup>[2]</sup>).
3. Compilar los módulos (recuerden que para poder compilar tienen que subir los makefiles <sup>[3]</sup> ya que no está permitido subir archivos binarios al repo).
4. Llenar archivos de config con la IP y el Puerto de cada módulo.

Una vez tengan todo listo, avísennos y comenzamos.

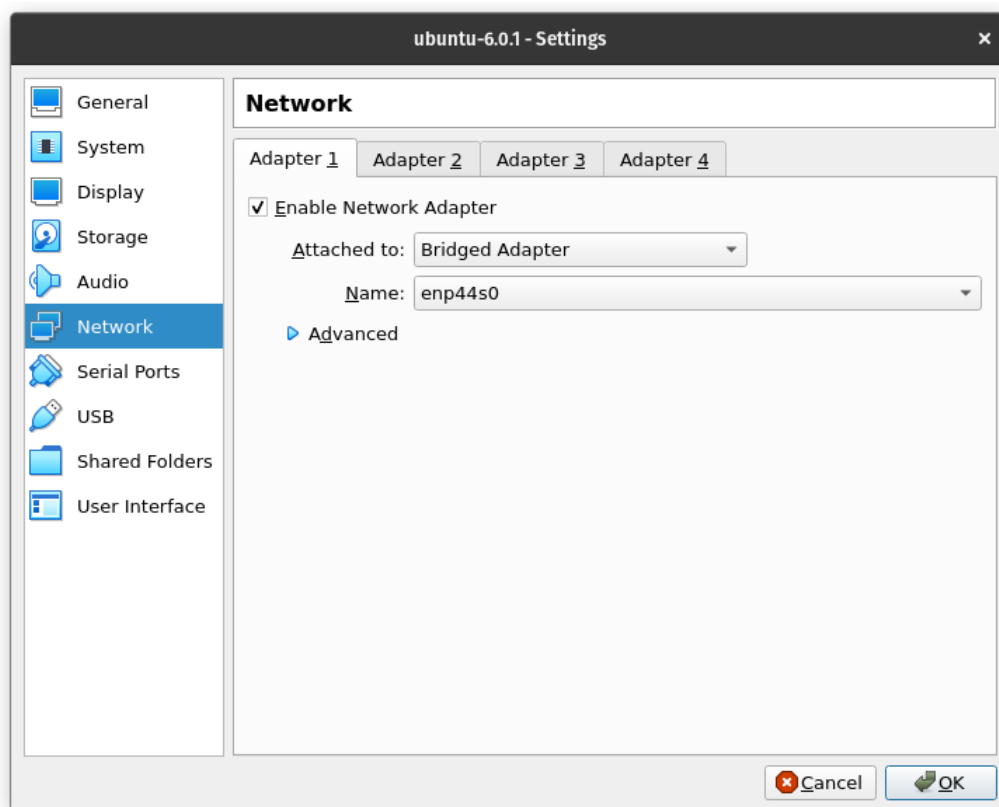
### TIP

Para ayudar un poco, les dejamos **algunos comandos que muy probablemente les van a servir**.

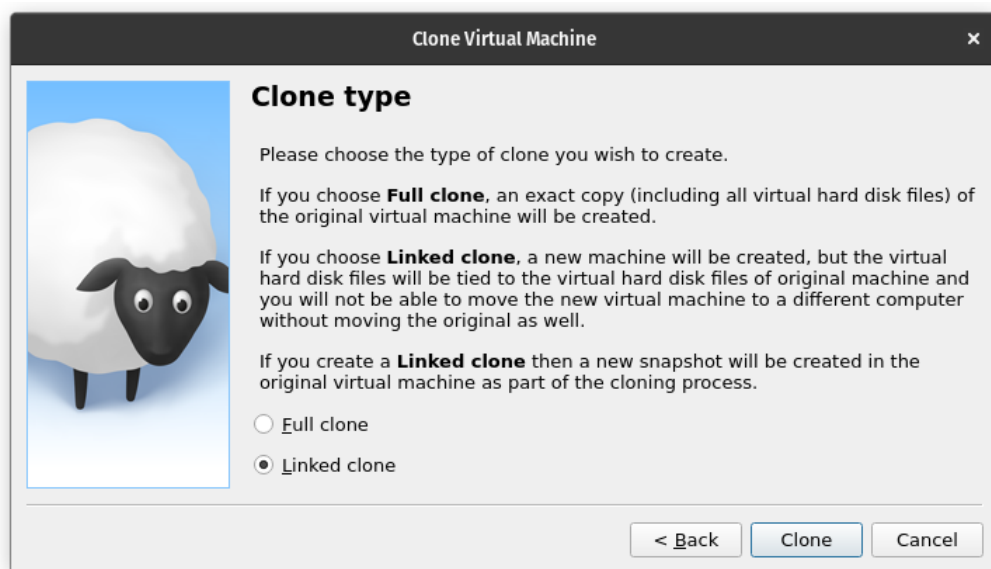
## Cómo practicar el despliegue en local

Desde sus computadoras, pueden armar una red local de VMs siguiendo estos pasos:

1. Ir a la configuración de la Ubuntu Server y en el apartado de "Network" elegir el "Bridged Adapter" (en español, "Adaptador Puente"):



2. Hacer click derecho en la VM > "Clone..." y seguir los pasos para clonar la VM. Recomendamos usar la opción "Linked clone" ya que el proceso de clonado es más rápido y ocupa menos espacio:



Una vez hecho esto, pueden usar el comando `ifconfig` para consultar la IP de ambas VMs y configurar los módulos.

### Paso 3: Día E (de entrega) - Obligatorio

A partir de este punto ustedes tendrán que seguir el instructivo de pruebas para reproducir lo que queremos evaluar, y mostrar a través de los logs que las cosas pasaron como tenían que pasar según la

prueba. También vamos a verificar que no haya esperas activas ni memory leaks usando http.

Queda a discreción de cada ayudante si hacer alguna modificación sobre una prueba si nota algún comportamiento extraño en su TP.

### Paso `segmentation fault (core dumped)`

En las sagas vikingas las armas se rompieron, A las misiones Apollo les costó diez intentos lograr un alunizaje. A Thomas Edison le costó más de mil intentos hacer una lámpara funcional, y a los hermanos Wright más de cuatro años (y varios huesos rotos) el primer avión. Nuestra mayor gloria no es no caer nunca, sino levantarnos cada vez que caemos (Confucio).



Cuando algo falla durante la entrega se reproduce un momento amargo para todos; y lamentablemente, debido a la cantidad usual de grupos, es difícil que podamos sentarnos con ustedes a revisar la causa del error. Sabemos que suena frío decir esto, pero también confiamos en que tienen el conocimiento y las herramientas para poder solucionar éste (y cualquier) problema que se presente.

Cuando esto suceda y tengamos que pedirles que nos cedan las máquinas, tomen 10 o 15 minutos para despejarse, y luego pónganse manos a la obra para solucionar este problema. Si todas esas figuras de la historia pudieron, ustedes también pueden tomarse unas horas, o incluso esperar a la próxima entrega para ver el problema más frescos. Cuando sientan que tienen todo en orden, pueden pedirnos volver a entregar, y si hay algún grupo que faltara pueden ocupar su espacio.

### Errores comunes

Para ayudar un poco, estos son algunos de los errores comunes que vemos en las entregas:

Error	Posible causa	Posible solución
En local mis módulos se conectan, pero en las VMs de prueba no.	Tener parametrizado estáticamente en el código (también conocido como <i>hardcodeado</i> ) <code>"localhost"</code> a la hora de conectar módulos.	Sacar el <code>"localhost"</code> y configurar <code>getaddrinfo</code> con los flags correctos.
Local funciona bien, pero en la entrega tengo comportamientos no determinísticos.	Local hay un solo CPU para todas las vms, pero en la entrega hay varios porque son varias máquinas. Ese paralelismo puede hacer que sus condiciones de carrera brillen más que con un solo CPU.	Probarlo distribuido con <b>Helgrind</b> (pueden levantar dos VMs server en sus máquinas si no pueden ir a Medrano) y usar semáforos.
Cuando bajo mi módulo servidor tengo que esperar 30 segundos porque sino no me andan los sockets.	Es una medida de seguridad de Linux para que no puedan robar paquetes enviados a tu servidor.	Configurar <code>setsockopt</code> para marcar a la IP y el puerto como reusables.
No puedo compilar mis módulos y no sé qué hacer.	Falta subir los makefiles del proyecto al repositorio o instalar la shared library.	Revisar el <b>paso 2</b> de esta guía.





Error	Posible causa	Posible solución
No tengo información suficiente para darme cuenta si la prueba anduvo o no.	No puse logs suficientes en mi TP porque creí que no era necesario.	<b>Agregarlos.</b> Sin logs no tenemos manera de saber que las cosas que tenían que pasar en la prueba pasaron. La foto final no garantiza que el trayecto haya sido el correcto.
Tardo demasiado en darme cuenta si la prueba anduvo o no.	Demasiado <b>printf debugging</b> 	Aprovechar que las commons proveen varios <b>niveles de logueo</b>  para mostrar solo la información relevante y ocultar los logs "de prueba" <b>sin comentarlos en el código.</b>

## Paso 4: Des-pliegue - Recomendado

¡Yastá! ¡Yastá! ¡Aprobaron la entrega! Pero antes de salir y contarles a sus seres queridos que están un paso más cerca de ser ingenieros, les pedimos por favor que se tomen la molestia de limpiar lo que hicieron. Borrar su repo de la server, cerrar sesión en cualquier cosa que hayan abierto y llevarse todo con lo que hayan entrado al labo. De esta manera, nos ahorramos todos un posible dolor de cabeza.

## Paso 5: Coloquio - Obligatorio

Una charla materia con dos ayudantes sobre cómo desarrollaron el TP a lo largo del cuatri y qué relaciones encontraron con la teoría.

1. Tener una fecha de expiración corta para nuestros tokens siempre es una buena práctica de seguridad. 
2. Si cuentan con una shared library propia, deberán copiar los headers (".h") en `/usr/local/include` y la biblioteca compilada (".so") en `/usr/local/lib` . Las commons se instalan **de la misma forma**    

3. En Eclipse, los makefiles son varios archivos que se encuentran en la subcarpeta `Debug/` y se llaman `makefile` o tienen la extensión ".mk". Para poder compilar es necesario subirlos **todos**. 