


<div style="text-align: center;"> Universidad Tecnológica Nacional Facultad Regional Avellaneda </div>										 Tecnatura Universitaria en Programación				
Materia:	Arquitectura y Sistemas Operativos ▾													
Pertenece a:	1º Cuatrimestre													
Apellido ⁽¹⁾ :					Fecha:									
Nombre/s ⁽¹⁾ :					Docente a cargo ⁽²⁾ :									
División ⁽¹⁾ :					Nota ⁽²⁾ :									
DNI ⁽¹⁾ :					Firma ⁽²⁾ :									
Instancia ⁽²⁾⁽³⁾ :	P1		RP1		P2		RP2		RIN	X	F			

(1) Campos a completar solo por el alumno.

(2) Campos a completar solo por el docente.

(3) Las instancias válidas son: 1º Parcial (**P1**), Recuperatorio de 1º Parcial (**RP1**), 2º Parcial (**P2**), Recuperatorio de 2º Parcial (**RP2**), Recuperatorio Integrador (**RIN**), Final (**F**), Recuperatorio de Final (**RF** - *Solo válido para seminario de nivelación*). Marcar lo que corresponda con una cruz.

Se contrató su talento en una empresa, para incorporarse al Área de DevOps.

Usted deberá demostrar sus conocimientos de SysAdmin Linux / Programador, al realizar las distintas tareas que se le piden.. y resolver los inconvenientes que puedan surgir en el camino, aplicando su expertise y buen criterio. (Demas está decir que en caso de reinicio, el servidor deberá levantar con todas las configuraciones aplicadas para poder brindar servicio sin realizar tareas previas.)

Precondiciones:

- **Crear un usuario con la 1ra letra del Nombre y el Apellido**
 - **Darle permisos de sudo (sin clave)** y agregar en los grupos que creas necesarios para realizar el examen.
- Tener git bash instalado en la vm y con su usuario ejecutar el siguiente comando
git clone https://github.com/upszot/UTN-FRA_SO_Examenes.git
- Prepararemos el historial... Ejecutar el siguiente comando
./UTN-FRA_SO_Examenes/202408/script_Precondicion.sh
- Una vez ejecutado el script de arriba ejecute:
source ~/.bashrc
- Tener en cuenta que Únicamente se evaluarán ejercicios resueltos por comando o scripting , NO se aceptarán ejercicios resueltos a través de interfaz gráfica.
- En el Home del usuario en la carpeta **RTA_Examen_\$(date +%Y%m%d)** se deberá dejar un archivo por cada punto de parcial con los comandos utilizados para realizar el mismo. Ejemplo: Punto_A.sh
- Dicha carpeta deberá subirse a un repositorio propio de git
"UTNFRA_SO_Recu_Integral_<Tu-Apellido>"
- **Todos los comandos se deben ejecutar desde el usuario del alumno.**
- En caso de requerir permisos de root, ejecutarlos con "sudo"

A. Partición-LVM:

La empresa compró los siguientes discos SSD (3GB,1GB,1GB) para incorporar el uno de los servidores productivos, La finalidad es alojar en dichos discos las imágenes de docker, la work-area de los usuarios de desarrollo y ampliar la memoria swap, Dejando parte de los discos libres para eventuales ampliaciones si así se requieren a futuro.

Los Montajes deben ser Permanentes: (Ver documentación de referencia en:

[fstab](#))

VG	LV	Tamaño	Punto de Montaje
vg_datos	lv_docker	10MB	/var/lib/docker/
vg_datos	lv_multimedia	1.5GB	/Multimedia/
vg_temp	lv_swap	2GB	Memoria Swap

Partición

Tamaño	Tipo	Punto de Montaje
512MB	Swap (82)	Memoria Swap

NOTA: Si ya contaba con docker corriendo en su Equipo, recuerde que después de montar el volumen, debe restartear el servicio de docker, para que se genere automáticamente la estructura de directorio.

```
sudo systemctl restart docker
sudo systemctl status docker
```

B. **Bash Scripting:** Se requiere la automatización mediante bash scripting el

Chequeo de URLs, cumpliendo las siguientes premisas:

Ubicación	<code>/usr/local/bin/<tu-apellido>_check_URL.sh</code>
Parámetros	<code><Path_Repo>/202408/bash_script/Lista_URL.txt</code>

- Tareas que debe cumplir:
 - Generar la siguiente estructura:

```
/tmp/head-check/  
├── Error  
│   ├── cliente  
│   └── servidor  
└── ok
```

- Verificar las URL del listado con el siguiente comando.

```
STATUS_CODE=$(curl -LI -o /dev/null -w '%{http_code}\n' -s "$URL")
```

- Registrar en `/var/log/status_URL.log` con el siguiente formato

```
20240707_225002 - Code:200 - URL:www.google.com  
20240707_225004 - Code:200 - URL:https://es.wikipedia.org/  
20240707_225005 - Code:500 - URL:https://httpbin.org/status/500  
20240707_225006 - Code:404 - URL:http://www.noexiste.com/  
20240707_225007 - Code:200 - URL:https://httpbin.org/status/200
```

- Dejar un archivo `<dominio>.log` con la misma línea en la carpeta según el siguiente cuadro.

STATUS_CODE	Carpeta
200	OK
de 400 a 499	cliente
de 500 a 599	servidor

```
/tmp/head-check/  
├── Error  
│   ├── cliente  
│   │   └── noexiste.log  
│   └── servidor  
│       └── httpbin.log  
└── ok  
    ├── google.log  
    ├── httpbin.log  
    └── wikipedia.log
```

C. Docker: Generar, y pushear en docker-hub una imagen de docker y correr la misma con **docker compose**.

Tareas a realizar:

Armar una Imagen:

- Trabajar dentro de la carpeta **<Path-Repo>/202408/docker/**
- Modificar el archivo **index.html** con los datos pedidos.

```
<li>Tu-Nombre Tu-Apellido</li>
<li>División: Tu-Division</li>
```

- Generar una imagen de docker Con el Nombre:
“**web3_RI2024-<Tu-Apellido>**” basada en nginx, con el contenido de la carpeta web provista, previamente modificado.
- Subir la misma a <https://hub.docker.com/>

Edite el archivo docker-compose.yml

- Reemplazar el contenido del archivo **web/file/info.txt** con la información de Modelo cpu y frecuencia en el siguiente formato.

```
CPU Modelo: i5-8250U Frecuencia: 1.60GHz
```

- Levantar la imagen pusheada (docker-hub):
 - Dejando en el puerto **8081** del host la página web levantada.
 - La carpeta **web/file** como volumen en el contenedor en el path: **/usr/share/nginx/html/file/**

D. Ansible:

- Playbook la ejecución de 4 roles con las tareas descriptas
- Se debe poder conectar por ssh simulando la conexión a otro servidor.
- Generar/Modificar 4 roles

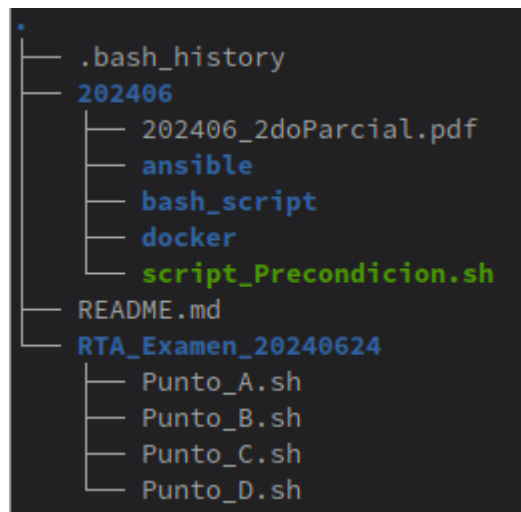
Nombre	Descripción de las Tareas que debe realizar
2PRecuperatorio	Dejar un archivo /tmp/alumno/datos.txt Con: Nombre: <Tu-Nombre> Apellido: <Tu-Apellido> División: <Tu-División> Fecha: <fecha del día> ----- Distribución: <Tu-Distro> Cantidad de Cores: <XX>
Alta_Usuarios_ <tu-apellido>	crear 2 usuarios: Profesor, alumno Agregarlos en los grupos, GProfesores, y GAlumnos según corresponda..
Sudoers_ <tu-apellido>	Agregar al grupo GProfesores para que pueda ejecutar sudo sin clave.
Instala-tools_ <tu-apellido>	instalar los siguientes programas. htop tmux tree speedtest-cli

E. **GIT:** Pushear en “**UTNFRA_SO_Recu_Integral_<Tu-Apellido>**”

tu repositorio creado de las premisas lo siguiente:

- Copiar la carpeta **<Path-Repo>/202408/** y todo su contenido.
- Copiar la carpeta **RTA_Examen_\$(date +%Y%m%d)** y todo su contenido
- **\$HOME/.bash_history** (Previa ejecución del comando “**history -a**”)

Nota: Debería de quedar algo similar a esto.



Formularios de Entrega:

https://docs.google.com/forms/d/e/1FAIpQLSevEzGMsdBpRD7Wz2NXTgyypfo93bOT0A_D4eckR5w8yW4zIQ/viewform