TSR LABS

COURSE 2020/21

DSIC'S LAB ENVIRONMENT

The goal of this document is to present sufficient detail about how the resources the DSIC offers for the labs are organized, so that students can better take advantage of them.

This is not an oficial DSIC document, and has been prepared as an additional tool for the students of TSR.

Resource virtualization and methods for remote accessing them are always useful tools. But in the present circumstances they become invaluable to allow access to resources without physical access to computers.

Unfortunately we should take care of other potential problems derived from potential low quality in the network connection a student has available. Check the following (https://virtual.blogs.upv.es/)

Note that the schedule and timing for accessing the lab resources are really the Schedule for technical support availability. There is the possibility that you can observe some malfunctions that you should try to resolve during the hours in which technical support is available.

Keep in mind that the likelihood of something not working as expected increases during university low activity or inactivity periods (weekends, vacations…).

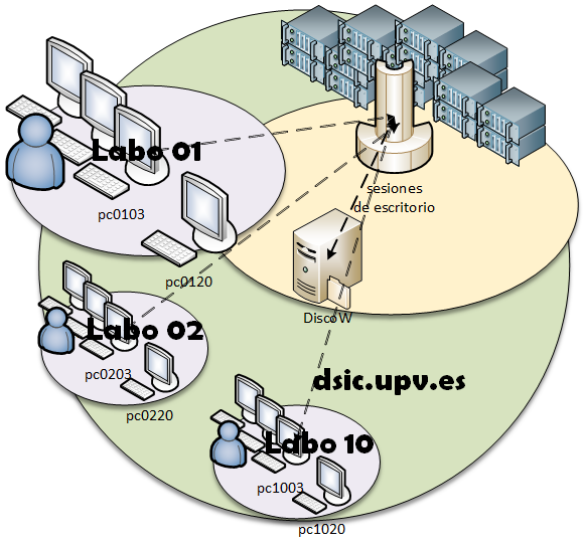
CONTENTS

The current physical lab setup is the result of more tan 30 years of evolution, where many different kinds of hardware/software has been put in productions, and where each new evolution neede to take into account a smooth transition between environments.

Recently, virtualization techniques are changing the way in which many organizations take advantage of their computing infrastructure. Such virtualization techniques provide many benefits in term of better utilization of the resources, as well as more convenience in accessing those resources, and allocating and apportioning them to users.

DSIC is one such organization that is taking advantage of these possibilities to set up an environment providing a high degree of flexibility for the docent needs of the students. IN fact one can say that the set of infrastrcutures provided by DSIC forma small CLOUD computing platform.

From this point of view, the actual computers available in the lab rooms are nothing more tan access points, which allow students to access the virtualized resources assigned to him within the DSIC cloud.



There are some exceptions to this usage of lab computers (when some courses have special requierements forcing those computers to boot into specialized environments/OSs). But the lab machine's most common usage mode is as access to the virtualized DSIC environment.

1 VIRTUALIZED RESOURCES

We have very briefly touched upon the advantages of virtualization, emphasizing the flexibility to allocate portions of the resources, and provide easy access to them.

When accessing a computing environment, a user may need to only perform actions that consume data stored in suitable storage medium, to produce results storable on some storage medium, but that do not alter the environment itself. For these cases, it may be sufficient to provide the user with some sort of **virtual desktop** environment from which to launch computations. When the result of such computations need to be persisted, a suitable persistent medium must be consistently available through the **virtual desktop** the user is provided with for each session. A typical way of doing it is through the usage of **remote directories** (e.g. folder DiscoW on Linux, or unit W: on Windows), that get attache to the environment set up for a user. Using remote directores offer the advantage of moving the **virtual desktop** session on any physical computer available.

However, there will be circumstances when the user needs to perform actions affecting the computing environment itself on which it is working (e.g., install new software, adding users, configuring a firewall, changing personal configuration files, etc…). In those cases, the user will need to have sufficient privileges to affect such state, and furthermore, the expectation is that those changes survive across different interaction sessions, for which they should be persisted. In such cases, the environment the user is provided resembles very much a complete computer, in fact being referred to as a Virtual Machine.

1.1 Virtual Desktops

Virtual desktops access a shared computing environment which does not produce complete isolation among users, producing some potentially undesirable effects:

- A remote desktop[1] is an interaction environment created within a session on a shared computer (shared with other virtual desktops). Many different users can have sessions on the same comupter. Whereas those sessions are sufficiently isolated to avoid users from modifying the characteristics of the environment, unfortunately they do not isolate the namespaces available among users: all userss see global namespaces for things like process IDs, users, or, more importantly, port numbers. This lack of isolation may provoke undesirable interference among users, especially in a lab. For instance, running who in linuxdesktop we can see the users logged in the system.

```
$ who
isaac_newton     pts/0        2018-09-18 19:39 (:12.0)
albert_einstein  pts/1        2018-09-11 10:34 (:25.0)
marie_curie      pts/2        2018-09-11 02:02 (:16.0)
nikola_tesla     pts/3        2018-09-13 12:25 (:21.0)
alan_turing      pts/4        2018-09-14 04:39 (:19.0)
ada_lovelace     pts/5        2018-09-18 15:40 (:04.0)
$
```

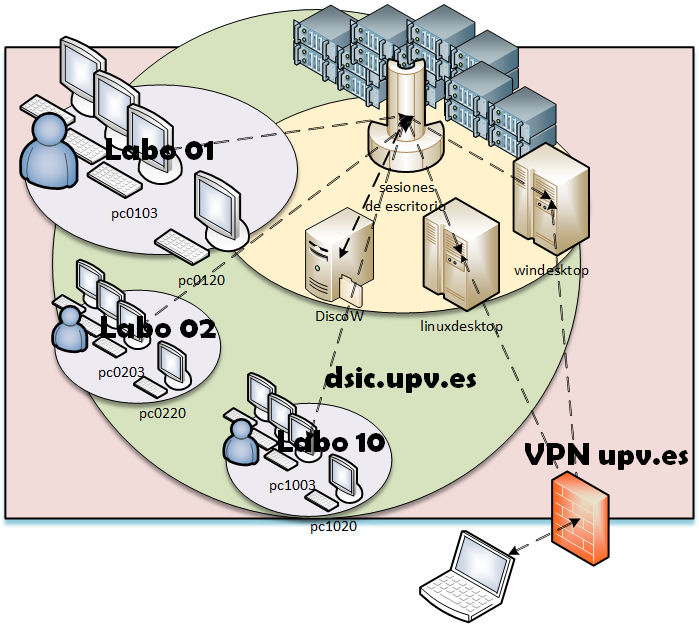
  The usual linux inspection tools would allow us to actuall list processes and ports being used by ALL users currently logged on the particular machine where the virtual desktop has been allocated.

- DiscoW does not behave like the "home" user directory. Most applications use some subfolder of $HOME to store their configuration for a user (e.g., VSC stores users-wide configuration within $HOME/.vscode). However, only DiscoW persists between sessions, not the whole $HOME directory. The effect is that changes in the configuration of those applications are not persisted.
  - On a virtual desktop, only DiscoW survives accross sessions. Any other change on the files outside DiscoW is lost, as the environment is reset to the same initial state each time a irtual desktop session is established for any user.
  - To sidestep the above, we suggest you create a folder ***myconfig*** within DiscoW whose content will be copied to $HOME when a session is started. Whenever we want to persist some configuration change, we need to explicitly move it to that directory. For instance, for VSC:

```
cp –pR $HOME/.vscode $HOME/DiscoW/myconfig
```

[1] Virtual desktops are no tour environment of choice for TSR given their lack of proper isolation, and insufficient privileges for their users.

Sessions on lab machines, and remote sessions to linuxdesktop and windesktop work this way.



1.2 Virtual Machines

When each user is provided exclusive usage of his own machine, his activity on that machine is fully isolated from the activity of the rest of their users on their own machines. Furthermore it is completely reasonable to provide users with full administrative privileges on their exclussively assigned machines, so that they can modify their configuration and persist it across sessions/boots.

Within a university environment, with thousands of students, it does not seem reasonable to offer an actual physical machine per student, with all the problems linked to it (need to provide support to resolve problems with that piece of equipment that may be used in circumstances out of the control of the support team).

However, as mentioned earlier, Machine Virtualization allows an organization (such as DSIC in our case) to actually offer not a physical machine per student, but a **Virtual Machine** per student. Management of the actual physical resources backing the virtualizaed machines can be streamlined by the support team iin an effective and efficient way.

Thus, each student receives a Virtual Machine (VM) he or she can access remotely through the network.

1. The student only has to worry about how to remotely access his/her VM.
2. Each VM is actully clone from a common template that can be customized for each teaching subject. If the student messes up the VM provided, it is always possible to return it to a pristine initial state by simply reinstalling this template (at the cost of losing whatever changes where persisted on the virtul disk of the virtual machine). Additionally, when a machine is not going to be used in a long time, it is possible to freeze it taking asnapshot of all its state, and storing it away. Later on, this snapshot can be used to construct another VM with the same state as the original one had when the snapshot was taken.

We will refer to these VMs as **portal VMs**[2].

In order to avoid accidental missuse by privileged systems users that are still learning how to properly use them, it is necessary to further isolate the networking environment where they operate.

To produce such isolation, access to these portal VMs must be carried out through its own isolated network, accesible trhough a portal. Access to this network must be carried out joining it through a VPN one must access expressely. This network joined through the special VPN is not connected to any other network, outside of this VPN you must establish (from your own desktop). Practically,

- Virtual machines within this "portal" receive a name within the cloud domain
- To ease access, all virtual desktop sessions can access the portal's VPN. If you establish a sesión on a labroom computer or windesktop y linuxdesktop, you can access the portal directly

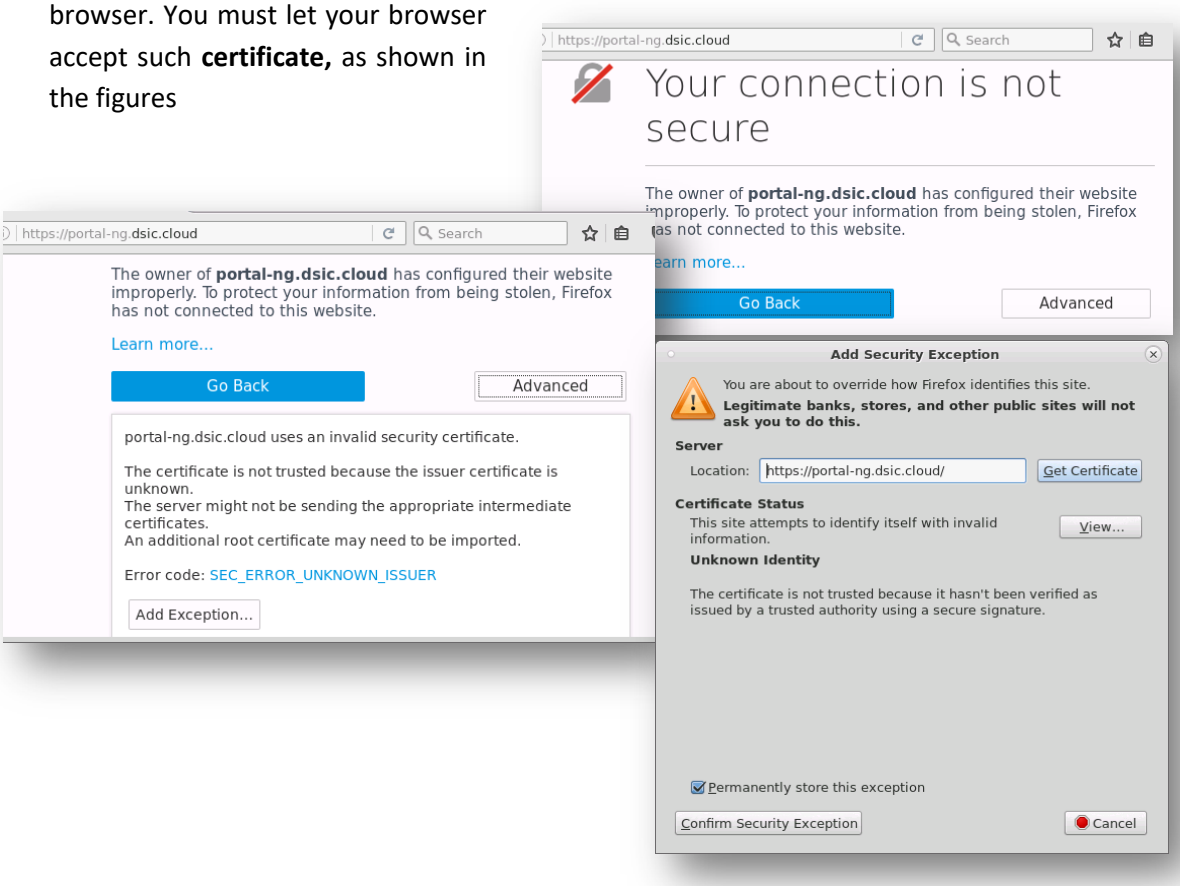
A VM can be in one of several states. In particular, it may be running ori t may be stopped.

To start them (e.g., boot them up) you must access server portal-ng.dsic.cloud, via a browser running on a machine connected to the portal's VPN.

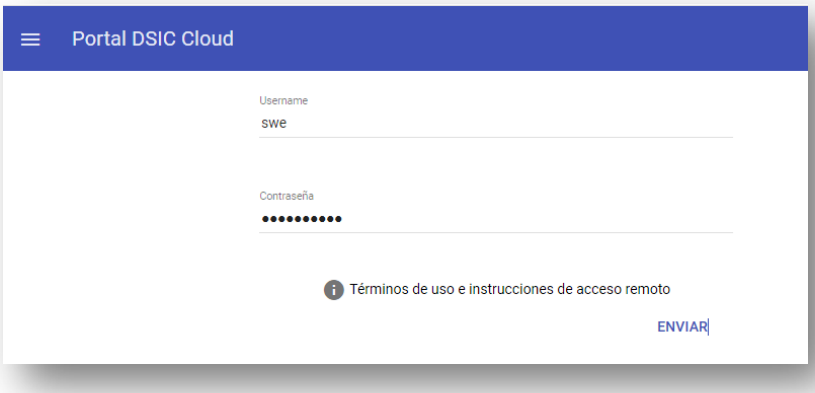
[2] Also referred to as Virtual Servers

When accessing portal-ng (**https://portal-ng.dsic.cloud**), you need to take into account the following issues:

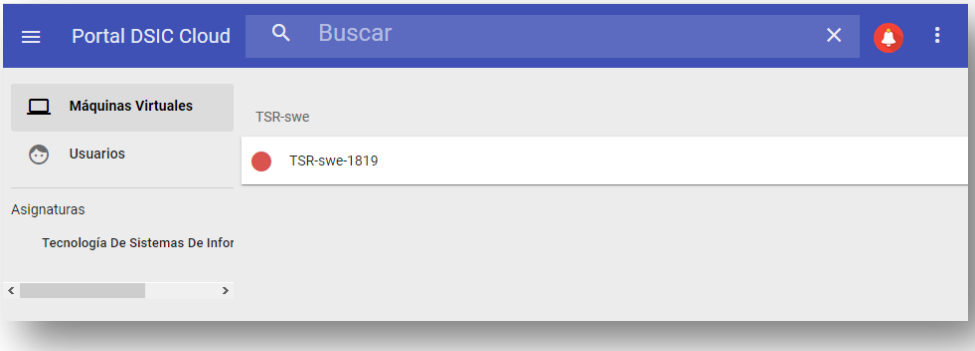
1. The first tiime you connect, your browser is going to complain about the site's certificate. This is due to the fact that it has not been sigen by an official cert authority known to your browser. You must let your browser accept such **certificate,** as shown in the figures



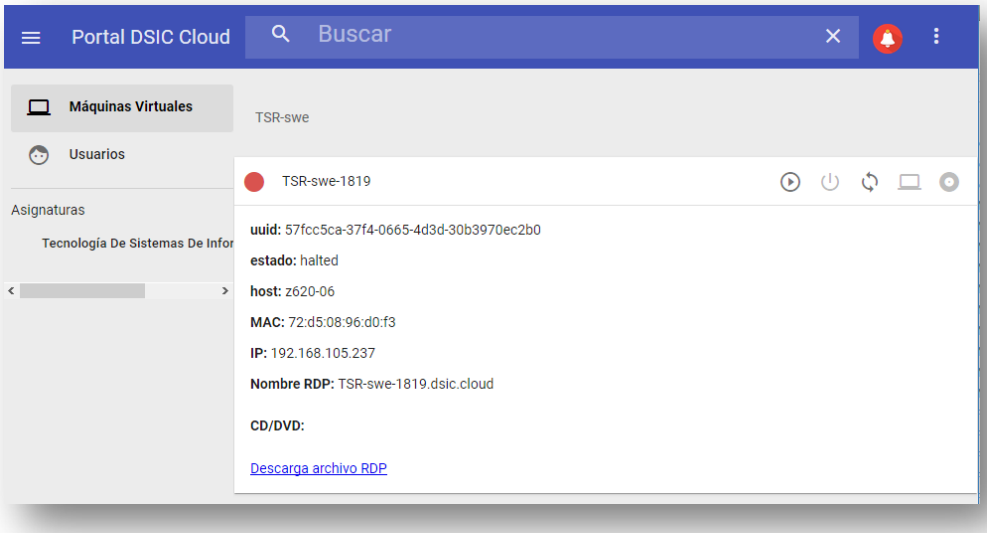
2. Once the browser has been configured as shown above, the web page will show a login dialog. The credentials you should use are those provided to you as a DSIC lab user.



3. Once you are successfully logged in, the portal shows the list of VMs the user has access to. The red dot indicates that the virtual machine is stopped. In the same row where we find that red dot, we can locate 4 icons that allow us to operate the VM: **start** ( ) the machine, **stop** ( ), **update** ( ) the information being shown or **initiate an interactive VNC session** ( ). A greyed-out icon indicates a non-available operation.



4. Clicking on the VM name shows us further details about the machine, including its FQDN and IP/MAC addresses.

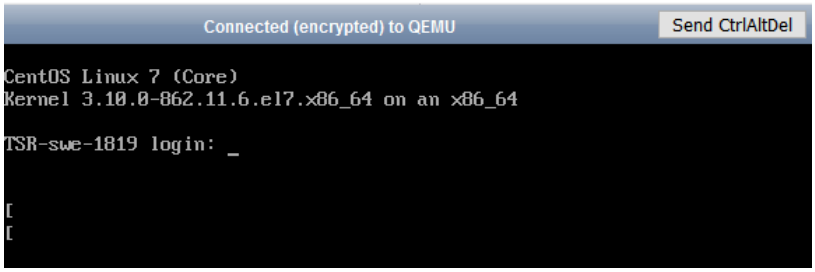


5. Once the machine has been started, we should expect to see the following:

   a) Notice that in this view, we get a shortened name for the machine. Its FQDN is shown in the detailed view of the machine as **Nombre RDP**.

b) When a VM is stopped, you whould use this interface to start it up. To stop a VM, however, it is best to do it orderly from within the VM itself, running shutdown –h now. Stopping your machine from the portal itself can be done, but its operation is equivalent to yanking the power cord from a computer[3]: it may corrupt its disk storage, making it challenging or even impossible to start it up again.

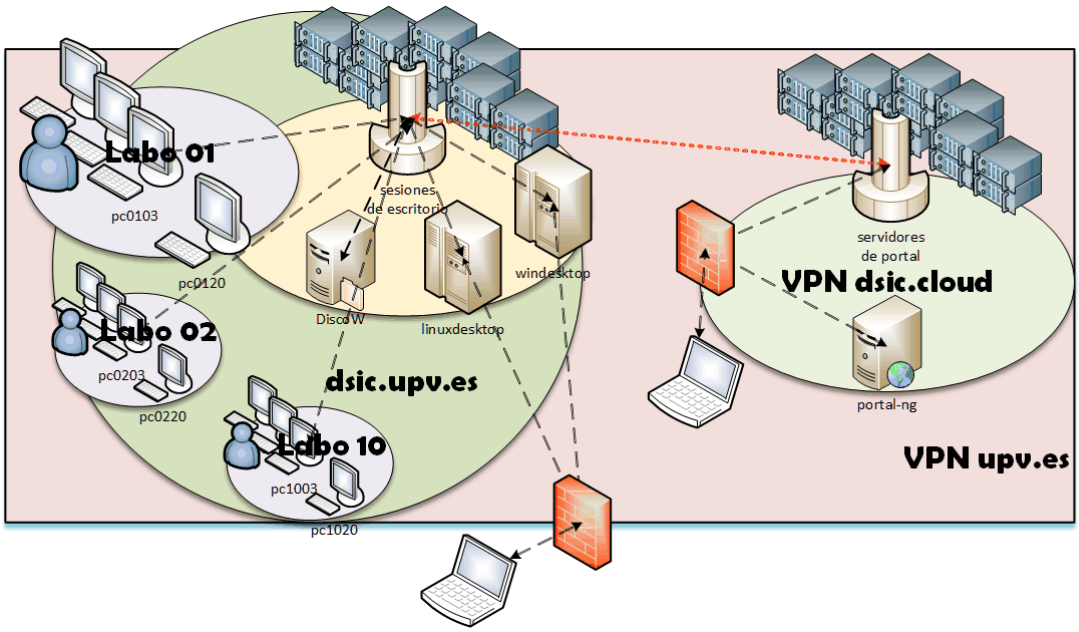
c) Accessing your virtual machine can be done through a VNC tty session, mediated by your browser, which will produce a certificate security warning like the one we mentioned earlier on. We do not advise accessing your VM this way, as the provided



interaction is quite **deficient**, and presents keyboard-related issues.

1.3 Summary

Feature	Virtual Desktop	Virtual Machine
Access to DiscoW (unidad W:)	Yes, within $HOME	No
Direct access from lab rooms	Yes	No
Similar to windesktop/linuxdesktop	Yes	No
Information outside DiscoW	Is lost	Is persisted
Can modify system configuration	No	Yes
Can modify app configuration	Yes, when implemented via files within $HOME, with special care.	Yes
Install/uninstall software	No, if using standard package management tools	Yes
Exclussive usage of resource	No	Yes
Internet access	Yes	Yes
Fixed preconfigured Linux/Windows version	Yes	No
Need to start/stop	No	Yes, through portal-ng



2 INTERACTING WITH THE PORTAL VM

We now explain some ways to interact with your assigned VM from your desktop machines. You will always need some network application on your desktop to connect to the VM.

In what follows we will use mivirtual as the name of your server (even so, some images may show the names of specific machines).

We will show how to carry out two kinds of interactions: **copy files** or start an interactive **remote session**.

2.1 Copy files

We can establish a console/tty sesión with the server using the sshd service available on the VM, through the ssh client program we should have on our desktop.

2.1.1 File copy through the command line

Using command scp it is possible to copy files from LINUX (UNIX) to other LINUX (UNIX) machines as long as the servers are running the sshd daemon (as is the case with our VMs)

- It is posible to use scp in Windows, by opening a Git-Bash console, and run scp within it.

```
scp archivos root@mivirtual.dsic.cloud:
```

Check scp man page for more detailed info on usage.

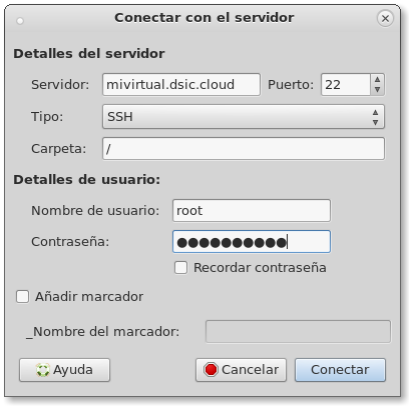
2.1.2 GUI-mediated file copying

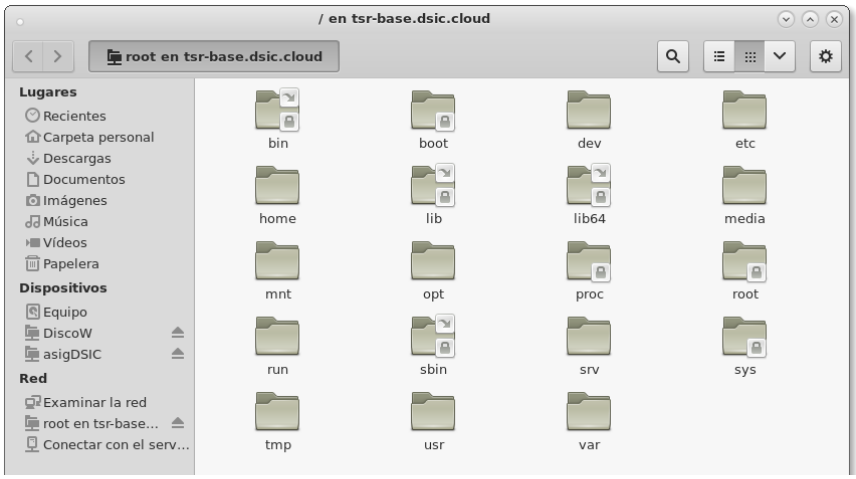
This operation is started from a LINUX client contacting LINUX servers, also leveraging the sshd daemon.

Within the System menú of your LINUX desktop GUI environment, you cn fin dan entry Conectar con el servidor…

- We can set it up to contact server mivirtual.dsic.cloud, with user root, port 22 and protocol ssh.

We will see a warning about the identity of the computer we are connecting to. We should discard the warning. Once this operation is carried out, we can work with the VMs file system through standard File Explorer Windows on our dektop.





You need to pay attention to the title and path show non top of the explorer windows to ensure your are copying to the right folders, and avoid mistakes.

In addition:

- Clicking on the background of a remote explorer window le tus start a terminal window connected to the VM via ssh.
- Local applications can work with remote files (they are "mounted" on the local system, as a remote disk).

In sum…

THIS MAY BE YOUR BEST OPTION, A IT ALLOWS SEAMLESS FILE TRANSFER AS WELL AS EASY EXECUTION OF REMOTE COMMANDS ON YOUR VM WHEN NEEDED.

2.2 Start a remote interactive session

2.2.1 Via a terminal window

(if you are using a Windows desktop, open a Git-Bash console to be ble to run ssh, scp).

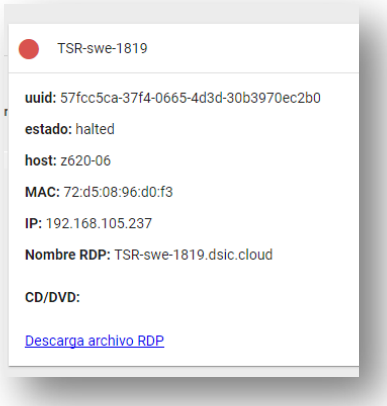
```
ssh root@mivirtual.dsic.cloud
```

Check the manpage for ssh for more detailed usage information.

2.2.2 Via upir desktop's GUI

Connect using Remote Desktop Connection protocol. This is supported the portal VM server environment, and is also supported from Linux and Windows client desktops.

- This mode of interaction has more overhead tan the one based on console terminals, but may be more intuitive to use.
- Through portal-ng we can find the configurtion parameters to establish an RDP session with our VM.
- If you have a Windows desktop, simply use the standard Conexión a Escritorio remote (Remote Desktop Connection) app.



If your desktop is LINUX:

- The Internet menu offers the Remmina app. We suggest for performance sake that you use a resolution of 1152x864 or 1024x768, and color depth of 16 bpp (*bits-por-pixel*).
- Other interfaces are base don rdesktop[4], but we advise using it through the command line:

```
rdesktop mivirtual.dsic.cloud -u usuario_remoto -a 16 -g '1024x768'
```

[4] Check rdesktop manual page for details on usage

- Another option is thinlinc from the graphics menu.

2.3 Closing details to take into account

All VMs share the same initial **root** password. Obviously you should change this password as soon as possible. Keeping the initial root password is a certain security hole and highly irresponsible. You are responsible for problems that can appear as a result of not taking care of changing that password.

Initial VM configuration has a firewall configuration that block external access to most VM ports. To enable communication to a range of ports (e.g., from 8000 to 9999), you need to configure the firewall like this:

```
firewall-cmd –permanent –add-port=8000-9999/tcp
firewall-cmd –reload
```

Given limited resources it may be necessary for two students to cooperate and share the same VM. This may lead to both concurrently interacting with the VM. It is important in these cases to organize access to this shared resources to avoid interfering with each other.

The following may be of help…

1. Verify if another user is connected to the VM when we log in.
2. Make use of some instant messaging/SMS/WA… to coordinate
3. Agree on a set of rules with the otehr user(s) to minimize chances of unwanted interference.

3 References

- Guide for student users of DSIC labs.(https://tracdsic.dsic.upv.es/depto/raw-attachment/wiki/LaboratoriosDocentes/Manual_del_usuario_labs-doc_V6_2016-09-08.pdf).
- Terms of use for the service for remote accessing virtual desktops and virtual machines at DSIC (http://www.upv.es/entidades/DSIC/infoweb/dsic/info/1043006normalc.html)
- PORTAL-NG student manual (http://www.dsic.upv.es/docs/infraestructura/portal-ng/manual_portal2_usuario_v8.pdf)