

Panell web de control de sistemes Cockpit

Introducció

Cockpit (<http://cockpit-project.org>) és una eina open-source d'administració de servidors que permet supervisar i administrar fàcilment sistemes Gnu/Linux individuals o múltiples a través d'un navegador web. Amb aquesta eina els administradors poden realitzar tasques de configuració de l'emmagatzematge o de les interfícies de xarxa, de control de serveis, de visualització de registres, de gestió d'usuaris, etc. També permet obrir un terminal dins del propi navegador com si fos una sessió oberta en SSH, entre moltes altres possibilitats.

La gran diferència de Cockpit respecte altres solucions més o menys similars és que no "inventa la roda". És a dir, no proporciona eines pròpies per realitzar les tasques d'administració del sistema sinó que aprofita les eines "estàndar" àmpliament conegudes que existeixen per cadascuna d'aquestes tasques (per exemple, per la gestió d'usuaris fa servir la suite "shadow" -comandes *useradd*, *usermod*, etc-, per la gestió de l'emmagatzematge fa servir la suite "storaged" -en parlarem més endavant-, per la gestió de la xarxa fa servir NetworkManager, etc). Per tant, Cockpit no és gens intrusiu perquè "només" funciona com una interfície web entre l'usuari i l'eina concreta a fer servir.

A Ubuntu tenim el paquet "cockpit" als seus repositoris habituals (*sudo apt install cockpit*) però no ve actualitzat a la seva darrera versió. A Fedora sí tenim la darrera versió als seus repositoris (*sudo dnf install cockpit*), així que serà preferible realitzar els exercicis corresponents en aquest sistema. Una altra diferència entre els dos sistemes està en la llista de "plugins" Cockpit que venen ja preinstal·lats o que estan disponibles als repositoris...un altre cop Fedora guanya perquè n'hi ha uns quants més que a Ubuntu (respecte quins "plugins" concrets hi ha a cada sistema i per a què serveixen cadascun d'ells en parlarem al proper apartat).

D'altra banda, cal fer notar que si instal·lem a Ubuntu el paquet Cockpit el servei corresponent es posarà en marxa just després de la instal·lació (i per tant ja estarà llest per ser utilitzat) mentre que si l'instal·lem a Fedora caldrà que iniciem i habilitem el servei explícitament (en aquest cas es faria així: *sudo systemctl enable cockpit.socket && sudo systemctl start cockpit.socket*).

NOTA: A Fedora, a més, cal vigilar que el tallafocs permeti el tràfic pel port 9090; això es pot aconseguir per exemple fent:
sudo firewall-cmd --add-port=9090/tcp && firewall-cmd --permanent --add-port=9090/tcp

Un cop instal·lat i comprovat que el "socket" *cockpit.socket* estigui encés (*systemctl status cockpit.socket*, veure més avall), podem obrir un navegador i accedir a la direcció <https://ipServidor:9090> (on, com podeu veure, el port d'escolta per defecte és el 9090). Hauria d'aparèixer (després del missatge d'error de "certificat autosignat", del qual ens oblidarem per ara) una finestra de login on es pot introduir qualsevol nom d'usuari (més la seva contrasenya) del sistema on estigui funcionant Cockpit. Si Cockpit detecta que l'usuari que inicia sessió a través d'ell pertany al grup "sudo/whell" (i, per tant, podria realitzar tasques d'administrador), Cockpit invocarà interna, automàtica i transparentment la comanda *sudo* (i sense que calgui introduir cap més contrasenya) cada cop que l'usuari, un cop loguejat, vulgui realitzar alguna acció amb privilegis d'administrador. En qualsevol cas, un cop iniciada la sessió, la interfície web és molt intuïtiva (s'explorarà als exercicis).

Estructura interna

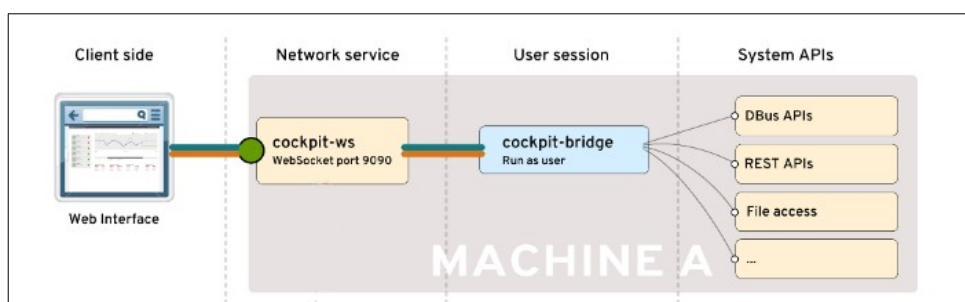
El paquet "cockpit" (d'Ubuntu o Fedora) incorpora implícites un parell de dependències indispensables que cal conèixer:

*Paquet "**cockpit-ws**": conté el servidor web, escoltant per defecte al port 9090, encarregat d'autenticar usuaris a través d'una pàgina web d'inici de sessió i, a partir d'aquí, de rebre les seves diverses accions des del navegador del client.

*Paquet "**cockpit-bridge**": l'encarregat de rebre de part de "cockpit-ws" les accions que ell al seu torn ha rebut dels usuaris i de cridar llavors a la comanda del sistema adient en cada cas per realitzar realment l'acció (segons el mètode que sigui necessari en cada cas: REST, D-Bus, etc). "Cockpit-bridge" és un procés que s'executa de forma permanent (a l'espera de rebre ordres) amb els mateixos privilegis que l'usuari indicat a la pàgina web d'inici de sessió (i si aquest pertany al grup "sudo/wheel", en el moment que ho necessiti "cockpit-bridge" invocarà a `sudo -o pkexec`, segons disponibilitat- automàticament)

NOTA: En realitat, el procés d'autenticació és realitzat per un procés anomenat *cockpit-session*, el qual és posat en marxa per *cockpit-ws* quan aquest rep un usuari i contrasenya. En aquell moment *cockpit-ws* passa aquestes dades a *cockpit-session* i és aquest qui comprova (via PAM o GSSAPI) si l'autenticació és exitosa. Si és el cas, ho notifica a *cockpit-ws* per a què aquest llavors posi en marxa *cockpit-bridge*.

En el següent diagrama es mostra gràficament aquesta divisió de tasques entre aquests components:



NOTA: Tot i que al diagrama anterior no es veu, també és possible accedir a altres servidors remots per administrar-los des d'un únic panell Cockpit.

A banda, amb la instal·lació estàndard del paquet "cockpit" en sistemes Ubuntu o Fedora s'instal·len altres paquets (no imprescindibles però sí molt convenients) que aporten, a mode de "plugins", "porcions de funcionalitat" que amplien la que de Cockpit ofereix per defecte. Podem destacar:

*Paquet "**cockpit-networkmanager**": permet gestionar les tarjes de xarxa de la màquina i la seva configuració (gràcies a l'API Dbus que proporciona el servei NetworkManager)

*Paquet "**cockpit-storage**": permet gestionar els dispositius d'emmagatzematge, incloent particions i punts de muntatge (gràcies a l'API Dbus que proporcionen els serveis Storage i Udisks)

*Paquet "**cockpit-system**": ofereix diferents utilitats bàsiques, com ara la gestió de serveis i la d'usuaris, la visualització de logs, l'obtenció d'informació bàsica del sistema, la monitorització bàsica de CPU o memòria, etc. També incorpora la funcionalitat de permetre obrir un terminal del sistema dins del panell web Cockpit

*Paquet "**cockpit-packagekit**": permet gestionar paquets (instal·lar, desinstal·lar, actualitzar, etc) des del panell web de Cockpit

Però, a més, Cockpit ofereix altres paquets, no instal·lats per defecte ni a Ubuntu ni a Fedora, que aporten encara més funcionalitat extra, com ara el paquet "**cockpit-podman**", el qual permet administrar contenidors Podman des de la interfície web de Cockpit i el paquet "**cockpit-machines**", el qual permet administrar-hi màquines virtuals de tipus Libvirt; el paquet "**cockpit-pcp**", el qual permet fer servir la suite PCP (<https://pcp.io>) per guardar les estadístiques d'ús de CPU, memòria, etc per a què es pugui consultar un històric; o el paquet "**cockpit-389-ds**", el qual permet administrar via web un eventual servidor 389DS que puguem tenir funcionant a la màquina.

NOTA: A Fedora a més tenim en exclusiva el paquet -no oficial- "**cockpit-file-sharing**", el qual permet administrar carpetes compartides SMB/NFS; el paquet -també no oficial- "**cockpit-navigator**", el qual ofereix les funcionalitats d'un gestor de fitxers (via web) en el servidor Cockpit, el paquet "**cockpit-selinux**", el qual permet, entre d'altres opcions, observar les alertes generades per SELinux o canviar el seu mode de "permissive" a "enforcing" i viceversa; el paquet "**cockpit-composer**", el qual permet generar isos de sistemes personalitzats via web fent servir tecnologia Kickstart; el paquet "**cockpit-sosreport**", el qual permet fer servir via web l'eina "sos" (<https://github.com/sosreport/sos>) per detectar i gestionar alertes de malfuncionament del sistema; i altres paquets més "nínxol" com "**cockpit-ostree**", "**cockpit-kdump**" i més. Si fos necessari algun d'aquests paquets extra concret a les classes ja s'indicarà en el seu moment.

"Socket" vs "Service"

Un aspecte molt interessant del Cockpit (més en concret, del servidor web "cockpit-ws") és que és un servidor que no està permanentment encès sinó que només arrenca "sota demanda" (és a dir, quan detecta una connexió externa). D'aquesta manera, no consumeix més recursos que els mínims imprescindibles, en el moment just. Per aconseguir això, el que passa és que sí que hi ha un component "escoltant" tota l'estona possibles intents de connexions, però aquest component no és pas el servidor *cockpit-ws* en sí sinó un "gos guardià" que només "despertarà el seu amo" (és a dir, encendrà el servei *cockpick-ws*) quan calgui. Aquest "gos guardià" és el que s'anomena, en el món Systemd, "socket". Per tant, si no hi ha cap client connectat ara mateix, el que hauríem de veure amb *systemctl status cockpit.socket* és que el "socket" de Cockpit està activat però el que hauríem de veure amb *systemctl status cockpit* és que el servidor no ho està. En canvi, si realitzem una connexió des d'un client (simplement mitjançant el navegador), veurem que automàticament el servidor s'haurà activat. Òbviament, si parem el "socket" de Cockpit (*systemctl stop cockpit.socket*) o el deshabilitem pel proper reinici (*systemctl disable cockpit.socket*) ja no hi haurà "gos guardià" amatent i, per tant, el servei ja no es podrà posar mai en marxa.

NOTA: El servidor "cockpit-ws" es posa en marxa quan el "socket" Cockpit detecta una connexió al port 9090 però ja hem dit que el procés "cockpit-bridge" només es posarà en marxa després d'haver-se loguejat un usuari vàlid. Aquest procés "cockpit-bridge" estarà funcionant fins que l'usuari tanqui sessió. D'altra banda, si durant els propers 10 minuts no hi ha cap intent de connexió més, llavors el servei "cockpit-ws" s'apagarà automàticament.

NOTA: De fet, "cockpit-ws" normalment s'executa darrere del proxy de finalització TLS "cockpit-tls" perquè només pot fer front per sí mateix a HTTP no xifrat. Per tant, el que s'activa en fer *systemctl start cockpit.socket* és el programa "cockpit-tls", que actua com un proxy HTTP de terminació TLS per a cockpit-ws. Més específicament, "cockpit-tls" gestiona un conjunt d'instàncies de cockpit-ws aïllades, una per certificat de client TLS, més una per a TLS sense certificat de client i una per a HTTP simple

Per canviar el port on escolta el "socket" Cockpit cal modificar la configuració del "socket" pròpiament dit i això no depèn de la configuració de Cockpit (ubicada a `/etc/cockpit/cockpit.conf`) sinó que depèn directament de Systemd. En general, la configuració de cada "socket" existent al sistema es troba en un determinat arxiu associat al servei corresponent, el qual té per extensió ".socket" i està ubicat dins de la carpeta `/usr/lib/systemd/system`. Concretament, es pot veure clar que la línia *ListenStream=* de l'arxiu `/usr/lib/systemd/system/cockpit.socket` és la línia que ens interessaria modificar per canviar el port d'escolta. No obstant, els arxius ".sockets" ubicats dins de `/usr/lib/...` no se solen modificar perquè provenen de la instal·lació per defecte de diferents paquets de la distribució i això vol dir que en qualsevol actualització podrien sobreescrir-se's (i, per tant, perdre's els canvis fets manualment). El que se sol fer llavors (aprofitant que a Systemd tenen prioritats sempre els arxius que hi ha dins de `/etc` sobre els arxius homònims que hi ha dins de `/usr/lib`) és crear dins de la carpeta `/etc/systemd/system` un fitxer ".socket" homònim a l'arxiu oficial ubicat a `/usr/lib/...` i afegir-li el mateix contingut que tingui aquest però canviant-hi tots aquells valors que, ara sí, desitgem. Un cop fet això, caldrà, no obstant, un últim pas, que és executar *systemctl disable nom.socket && systemctl enable nom.socket* (o bé, més fàcil: *systemctl daemon-reload*) per a què el canvi d'ubicació de la configuració es tingui en compte a partir d'aquest moment.

Conclusió: el que hem de fer per a què Cockpit escolti en un altre port és copiar el fitxer `/usr/lib/systemd/system/cockpit.socket` a la carpeta `/etc/systemd/system` i editar la línia *ListenStream=* d'aquest nou fitxer per tal d'indicar el número de port que volguem. Per comprovar que s'apliqui la configuració personalitzada, podem executar *systemctl cat cockpit.socket* (ens ensenyarà la configuració actualment activa) o també veure directament la sortida de *systemctl status cockpit.socket*

NOTA: En compte d'utilitzar l'arxiu `/etc/systemd/system/nom.socket` hi ha una manera alternativa d'indicar configuracions personalitzades, la qual té l'avantatge de no haver d'especificar totes les línies d'un fitxer ".socket" estàndard sencer sino només les línies de les seccions que volem que siguin diferents respecte el fitxer oficial `/usr/lib/systemd/system/nom.socket`, el qual se segurà tenir en compte per les línies sense modificar. Aquest mètode consisteix en crear una carpeta anomenada `/etc/systemd/system/nom.socket.d` i allà dins crear un fitxer anomenat "nom.conf" contenint només les línies de les seccions que volem que siguin diferents a les que hi ha a l'arxiu `/usr/lib/...` Per exemple, per canviar el port d'escolta del Cockpit es podria crear un arxiu anomenat `/etc/systemd/system/cockpit.socket.d/cockpit.conf` amb el següent contingut (i seguidament executar la comanda *systemctl daemon-reload*):

```
[Socket]
ListenStream= #Aquesta línia és necessària per "treure" el port original indicat a "/usr/lib/..."
               #perquè si no, s'acumula (ja que es tenen en compte els dos fitxers)
ListenStream=9898 #Aquesta línia és la que afegeix el nou port
```

Cal fer notar que el procediment anterior, en realitat, s'hauria pogut fer de forma més senzilla simplement executant la comanda `systemctl edit cockpit.socket` i escrivint les línies anteriors a l'editor que hi apareix i guardar.

NOTA: En sistemes Fedora és important tenir en compte que si SELinux està en mode "enforcing" no ens deixarà fer el canvi de port (o millor dit, quan volguem iniciar el socket Cockpit ens donarà un error, que es pot consultar al Journal). La solució podria ser etiquetar el nou port per tal de què tingui l'etiqueta "websm_port_t" (veieu `sudo semanage port -l | grep 9090`) executant aquesta comanda: `sudo semanage port -m -t websm_port_t -p tcp 9898` (si el port 9898 no estigués associat a cap etiqueta, s'hauria d'escriure el paràmetre `-a`, d'"afegir", en comptes de `-m`, de "modificar") o, més fàcil, posar el SELinux en mode "permissive" (això ho podem fer temporalment així: `sudo setenforce permissive`)

HTTPS

Cockpit escolta connexions tant HTTP com HTTPS a través del/s port/s indicats mitjançant la directiva `ListenStream=` ja coneguda. Si es detecta una connexió HTTP, Cockpit redirecciona automàticament el client a una connexió HTTPS (amb 2 excepcions: si la connexió prové de 127.0.0.0/8 o si demana la URL `"/ping"`).

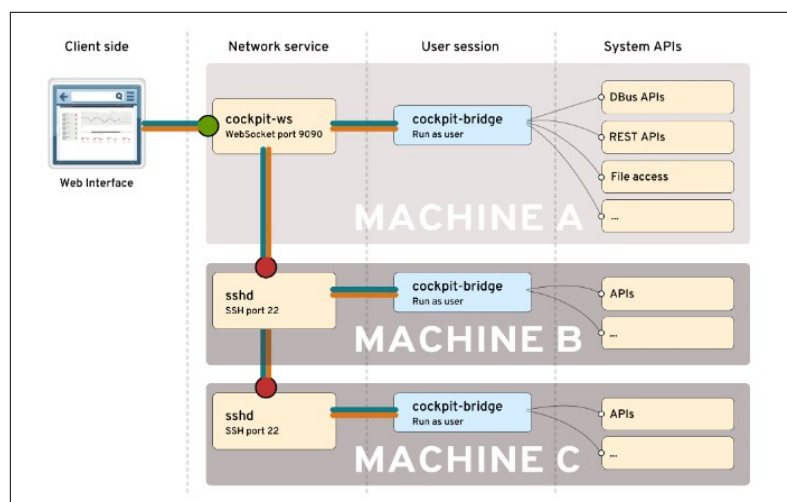
En general, però, per poder implementar un servidor HTTPS és necessari que el servidor tingui accés a un fitxer especial anomenat "certificat de servidor". El/s certificat/s de servidor que utilitza Cockpit per poder atendre connexions HTTPS (tots ells fitxers amb el format PEM i amb l'extensió "cert" o "crt") s'ha/n d'ubicar dins de la carpeta `"/etc/cockpit/ws-certs.d"`. Si no hi hagués cap certificat en aquella carpeta, la primera vegada que un client qualsevol connecti a través de HTTPS amb Cockpit, aquest utilitzarà automàticament la suite OpenSSL per generar un certificat autosignat amb el nom de **"0-self-signed.cert"** (el qual inclou dins seu també la clau privada associada, en un bloc distingit pel títol BEGIN PRIVATE KEY) i farà servir aquest a partir de llavors a totes les connexions HTTPS posteriors. És per això que la primera connexió a un servidor Cockpit recentment instal·lat "falla": perquè encara no s'ha creat cap "certificat de servidor".

NOTA: Per evitar aquesta "fallada", a la mateixa carpeta on es troba el certificat "0-self-signed.cert" s'haurà generat un altre fitxer, anomenat **"0-self-signed-ca.pem"** que representa el certificat arrel de la eventual CA que ha signat el certificat autosignat. Si aquest certificat arrel s'importa prèviament a la configuració dels navegadors clients (al Firefox es faria així: *Preferències -> Privadesa i seguretat -Mostra els certificats -> Importa*) això farà llavors que aquests navegadors ja "confiïn" en la validesa del certificat autosignat i, per tant, ja no aparegui la pantalla de "fallada"

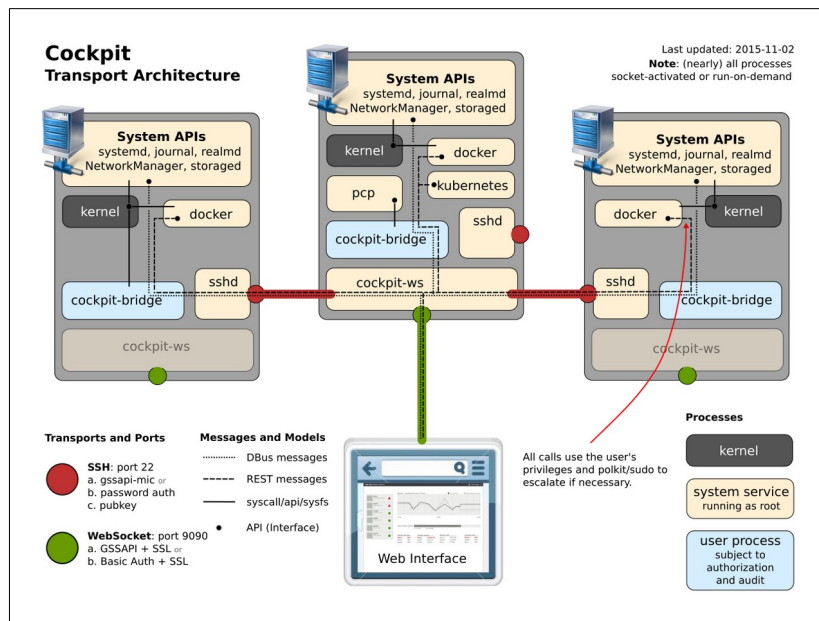
Si hi hagués més d'un certificat en la carpeta, Cockpit utilitzarà només l'últim (ordenant-se per ordre numérico-alfabètic segons el seu nom de fitxer). Per saber amb quin certificat concret Cockpit treballa, es pot executar la comanda `sudo remotectl certificate`

Administració de servidors remots des de Cockpit

Una opció interessant de Cockpit que hem comentat abans de passada és la possibilitat de poder administrar servidors remots "finals" a partir d'un servidor Cockpit "principal". Per aconseguir això, el servidor "final" només cal que tingui un servidor SSH en marxa i el paquet "cockpit-bridge" instal·lat (a més de qualsevol altre subpaquet/plugín que permeti interaccionar amb el sistema, com ara "cockpit-system", per exemple), i ja està. El següent esquema mostra aquesta idea:



Com es pot veure a l'esquema anterior, el servidor final no cal que tingui cap "socket" obert perquè, de fet, no tindrà cap servidor "cockpit-ws" que encendre. Un diagrama similar però més complet és aquest:



El procediment bàsicament és el següent: un cop iniciada la sessió en el panell web del servidor Cockpit "principal" (i sempre que l'usuari pugui realitzar tasques d'administrador, no val un usuari "pelat"), s'ha de pulsar sobre la fletxeta apuntant cap avall que apareix a la part superior de barra lateral esquerra, al costat del nom del servidor on estem connectats; en fer-ho apareixerà un botó "Add new host" que servirà per afegir el servidor "final" que es vol monitoritzar/administrar (es preguntarà la direcció IP d'aquest servidor remot, si és la primera vegada, la clau de màquina SSH a acceptar, i l'usuari/contrasenya que es farà servir per entrar en aquella màquina remota -alternativament es podria fer ús de claus públiques/privades SSH-). A partir d'aquí, es podrà anar alternant entre els panells de configuració web corresponents a les diferents màquines "reconegudes" simplement pulsant sobre el desplegable que apareixerà a la part superior de la barra lateral esquerra.

Una representació dels servidors "finals" reconeguts pel servidor Cockpit es guarda dins de la carpeta `"/etc/cockpit/machines.d"` en forma d'arxius JSON. Si hi hagués més d'un fitxer (no és habitual), l'últim sobreescrui als anteriors (el seu nom s'ordena numéricoalfabèticament). Cadascun d'aquests fitxers conté un objecte que mapeja l'identificador de la màquina (que podria ser la seva IP o bé qualsevol altre valor únic) a objectes que defineixen les propietats d'aquesta màquina com ara "visible", "color" (en format `rgb(r,g,b)` on `r,g` i `b` són valors entre 0 i 255), "address", "user" (per indicar si s'usa un altre usuari diferent de l'emprat en entrar al panell web per loguejar-se a aquest servidor final), "port" (per indicar el port SSH on està escoltant el servidor final...per defecte és 22), etc.

Arxiu de configuració `"/etc/cockpit/cockpit.conf"`

L'arxiu de configuració general de Cockpit és `"/etc/cockpit/cockpit.conf"`. Aquest fitxer té una sintaxis de tipus INI (és a dir, conté parelles de clau->valor agrupades per seccions). Un cop s'hagi gravat el/s canvi/s que es realitzi/n en aquest fitxer, per a que s'apliqui efectivament caldrà reiniciar el socket Cockpit (`sudo systemctl restart cockpit.socket`). D'entre les directives més interessants (la llista completa es pot trobar a man `cockpit.conf`) podem destacar:

*A la secció [WebService] :

**LoginTitle* : Estableix el títol que mostrarà el navegador a la pantalla d'inici de sessió

**LoginTo* : Si s'estableix a *false*, l'opció de "Connect to" de la pantalla d'inici de sessió que permet connectar a altres servidors Cockpits remots desapareixerà

**MaxStartups* : Estableix el número màxim d'inicis de sessió concurrents permesos. Per defecte val 10

* *Shell* : Estableix la ruta (relativa a partir de la carpeta "/usr/share/cockpit") del component de Cockpit (la "pàgina web") que es mostrarà un cop iniciada la sessió. Per defecte val "/shell/index.html".

*A la secció [Session] :

**Banner* : Indica la ruta d'un fitxer el contingut del qual es mostrarà a la pantalla d'inici de sessió. Normalment es fa servir l'arxiu "/etc/issue" però per defecte aquesta directiva no té assignat cap valor

**IdleTimeout* : Indica el temps (en minuts) després dels quals la sessió oberta es tancarà si no hi ha activitat. Per desactivar aquest logout automàtic cal assignar a aquesta directiva el valor 0. Per defecte val 15. Noteu, però, que aquesta directiva només funciona en el cas de realitzar inicis de sessió via contrasenya: si es fan servir altres mètodes com ara Kerberos, OAuth o amb certificats, el logout automàtic no és possible.

NOTA: El port d'escolta del servidor Cockpit no es canvia en aquest fitxer, recordeu: cal fer-ho a l'arxiu "cockpit.socket" de Systemd

EXERCICIS:

1.-Arrenca una màquina virtual Fedora que tingui la seva tarja de xarxa en mode adaptador pont i un usuari (l'anomenarem "usuari") amb la possibilitat d'executar *sudo*. Si el sistema Fedora és de tipus Server, Cockpit ja vindrà instal·lat; si és Workstation l'hauràs d'instal·lar. Comprova que el "socket" estigui iniciat (el "service" no!) i que el port 9090 no estigui bloquejat per cap tallafocs. A partir d'aquí:

a) Vés a la secció "Overview" i descriu breument quina informació et mostren els diferents apartats que allà apareixen ("Health", "Usage", "System information" i "Configuration")

b) Vés a l'enllaç "View details & history" que apareix a la secció "Usage" del panell "Overview" ¿Què veus?

c) Vés a l'enllaç "View hardware details" que apareix a la secció "System information" del panell "Overview" ¿Què veus?

d) ¿Què passa si pulses al botó "Limited access" que apareix a la cantonada superior dreta del panell web (o sobre l'enllaç "Turn on administrative access" que apareix a la barra groga superior) i escrius la contrasenya de l'usuari "usuari"? ¿Què hauries de fer per tornar a l'estat original?

e) Comprova com, amb l'usuari amb privilegis pots canviar el nom de la màquina i també la seva data, hora i zona horària clicant sobre els enllaços pertinents de la secció "Overview" i aturar/reiniciar el sistema clicant sobre el desplegable pertinent però amb l'usuari sense privilegis aquests enllaços o desplegable no apareixen.

2.-a) Amb l'usuari amb privilegis, vés a la secció "Logs" i observa tota la informació que t'ofereix del errors (o missatges pitjors) només en el dia d'avui. ¿Quin és el programa que causa més errors?

b) Vés a la secció "Storage" i observa tota la informació que t'ofereix. ¿Què mostren les gràfiques "Reading" i "Writing"? ¿I la taula "FileSystems"? ¿I la taula "Drives"? ¿Pots refer la taula de particions des d'allí? ¿Des d'on pots esborrar una determinada partició? ¿I reformatejar-la? ¿I desmuntar-la?

c) Vés a la secció "Networking" i observa tota la informació que t'ofereix. ¿Què mostren les gràfiques "Transmitting" i "Receiving"? ¿On pots definir la IP/màscara/porta d'enllaç/Servidors DNS de les tarjes de xarxa mostrades a la taula "Interfaces"? ¿Per a què creus que serveix la taula "Firewall" (si està activada; per a què ho estigui cal que prèviament el servei *firewalld* ha d'estar encés)?

NOTA: Les opcions de "Add bond", "Add bridge" i "Add VLAN" les estudiarem a Xarxes més endavant

NOTA: En el cas de què el servidor Cockpit estigui funcionant en un Ubuntu Server és possible que aparegui el missatge "Unmanaged interfaces". En aquest cas, cal realitzar els següents passos addicionals (amb els quals s'estarà indicant, que en lloc de fer servir Systemd-Networkd es faci servir NetworkManager com a gestor de xarxes per defecte del sistema):

```
mv /etc/netplan/01-netcfg.yaml /etc/netplan/01-netcfg.yaml.bak
echo -e "[keyfile]\nunmanaged-devices=none" > /etc/NetworkManager/conf.d/10-globally-managed-devices.conf
systemctl restart NetworkManager
```

d) Vés a la secció "Accounts" i observa tota la informació que t'ofereix. ¿Des d'on pots bloquejar un determinat usuari? ¿I canviar-li la contrasenya? ¿I fer que pugui executar *sudo/pexec* ? ¿I tancar-li la sessió? ¿I veure l'historial dels inicis de sessió? ¿Quines d'aquestes opcions deixen d'estar disponibles si l'usuari no té privilegis?

dII) ¿A quina comanda (relacionada amb SSH) seria equivalent el botó "Add Key" que apareix a l'apartat "Authorized public SSH keys" de la pàgina corresponent a un usuari qualsevol dins de secció "Accounts" de Cockpit? Per tant, ¿què s'aconsegueix si s'usa aquest botó adientment?

e) Vés a la secció "Services" i observa tota la informació que t'ofereix. ¿Què significa el quadre "System| User" que apareix a la cantonada superior dreta de la finestra principal d'aquesta secció? ¿Des d'on pots iniciar/aturar un servei/timer/socket/... concret? ¿I habilitar-lo/deshabilitar-lo? ¿Què mostra el desplegable "Show relationships" que apareix a la finestra de detalls d'un servei/timer/socket/... qualsevol?

f) Vés a la secció "Software updates" i observa tota la informació que t'ofereix. ¿Des d'on pots comprovar si hi ha actualitzacions disponibles? ¿I activar/desactivar les actualitzacions automàtiques? ¿I fer només les actualitzacions de seguretat?

g) Vés a la secció "Terminal" i observa tota la informació que t'ofereix

h) Vés a la secció "Applications" i instal·la el complement "Podman" (el farem servir després)

NOTA: En aquesta secció apareixen els paquets oficials de Cockpit que amplien la seva funcionalitat bàsica (estan mencionats a la teoria), els quals s'anomenen "applications" (podríem entendre'ls com "mòduls" o "plugins". Aquestes "applications" també es podrien haver instal·lat (com, de fet, passa amb les no oficials, tal com veurem), directament des del repositori de paquets de la distribució amb l'eina *dnf*

3.- Canvia el port d'escolta de Cockpit per a que sigui el 9898 fent servir la comanda *sudo systemctl edit cockpit.socket* ¿Quin fitxer apareix llavors a la línia "Loaded" en fer *systemctl status cockpit.socket* ? ¿I a la nova línia "Drop-in"? ¿I a la línia "Listen"?

4.-a) Esborra el certificat HTTPS que genera Cockpit per defecte i intenta entrar amb el navegador al servidor Cockpit un altre cop. ¿Quina pàgina web surt i per què creus que passa això? ¿Què veus de nou ara a l'interior de la carpeta "/etc/cockpit/ws-certs.d"?

b) Vés seguidament a l'apartat del Firefox "Preferències"->"Privadesa i seguretat"->"Certificats"->"Mostra els certificats"->"Servidors" ; què veus allà? ¿Què passa si esborres de la llista que et mostra el Firefox el certificat corresponent al teu servidor Cockpit i tornes a intentar accedir-hi?

5.- a) Crea una nova màquina virtual (de tipus Server, però és igual si Ubuntu o Fedora) amb la seva tarja de xarxa també en mode "Adaptador pont" i instal·la-hi només els paquets "cockpit-bridge" i "openssh-server"

NOTA: Si, per anar més ràpid, clones la màquina on ara mateix tens instal·lat el Cockpit, tingues en compte que a la màquina clon hauràs de desinstal·lar-lo primer per poder fer l'exercici correctament. És a dir, abans d'instal·lar els dos paquets indicats hauràs de fer el següent (suposant un Ubuntu): *sudo apt purge cockpit cockpit-bridge && sudo apt autoremove*

b) Canvia el nom d'aquesta nova màquina a "final" (*hostnamectl set-hostname "final"*) i fixa't en la seva IP

c) Inicia sessió (amb privilegis) en el panell web del servidor Cockpit "original" i pulsa sobre la fletxeta apuntant cap avall que apareix a la part superior de barra lateral esquerra, al costat del nom del servidor on estem connectats; en fer-ho apareixerà un botó "Add new host". Utilitza'l per afegir el servidor "final" creada al primer apartat: d'aquesta manera podràs monitoritzar/administrar-la (es preguntarà la seva adreça IP i, si és la primera vegada, la clau de màquina SSH a acceptar, i l'usuari/contrasenya que es farà servir per entrar en aquella màquina remota -alternativament es podria fer ús de claus públiques/privades SSH-). ¿Què veus?

NOTA: A partir d'aquí, podràs alternar entre els panells de configuració web corresponents a les diferents màquines "reconegudes" simplement pulsant sobre el desplegable que apareixerà a la part superior de la barra lateral esquerra.

d) Instal·la a la màquina "final" el paquet "cockpit-system" i surt i torna a entrar a la sessió web de Cockpit. ¿Què veus ara?

e) Torna a sortir de la sessió web de Cockpit i ara, a l'apartat "Other options"->"Connect to" de la pantalla d'inici de sessió escriu la IP de la màquina "final" (i un usuari i contrasenya vàlids per aquella màquina). ¿Què veus quan intentes entrar? Finalment, apaga aquest servidor final...no el farem servir més.

6.-a) Després de llegir <https://github.com/cockpit-project/cockpit/blob/main/doc/branding.md>, canvia el logo i el fons de la pantalla d'inici de sessió web de Cockpit

b) Escriu alguna cosa dins del fitxer `"/etc/motd"` i tanca sessió al panell web de Cockpit. ¿Què veus en tornar-hi a entrar?

c) ¿Què passa si afegeixes a l'arxiu `"/etc/cockpit/cockpit.conf"` (que pot no existir) les següents línies, reinicies el servei Cockpit i tornes a iniciar sessió? ¿Per què?

```
[WebService]
Shell = /system/terminal.html
LoginTo=false
[Session]
IdleTimeout=1
```

7.-a) Inicia el socket "podman" de la manera que vulguis (via Cockpit, via terminal...) per tal d'oferir la possibilitat a programes de tercers (com és Cockpit) de comunicar-se amb Podman i així poder usar les seves funcionalitats des d'aquests programes de tercers. Un cop fet això, vés a la secció "Podman containers" i:

* Utilitza l'opció "Download new image" del desplegable de la zona superior dreta per tal de descarregar la imatge (propietat de l'usuari, no del sistema!) anomenada "docker.io/ubuntu/apache2"

* Crea, a partir de la imatge anterior, un contenidor (propietat de l'usuari, no del sistema!) amb totes les seves opcions amb el valor per defecte excepte que cal que afegeixis (sota l'apartat "Integration") un "Port mapping" de tipus 80->80. Procura que cap programa del sistema amfitrió estigui utilitzant el port 80 ara mateix.

* Posa en marxa el contenidor anterior i comprova que estigui funcionant anant al navegador de la màquina real i escrivint l'adreça de la màquina virtual: haurà d'aparèixer la pàgina web per defecte del servidor Apache, accessible a través del mapeig de ports realitzat. ¿Quina informació mostra l'apartat "Details" del contenidor en qüestió

* Finalment, atura el contenidor, elimina'l i esborra també la imatge associada

b) Instal·la el paquet "cockpit-navigator" i reinicia sessió en el Cockpit. ¿Quina secció nova veus al menú lateral esquerra? ¿Què s'hi veu si hi accedeixes? ¿Quines accions pots realitzar mitjançant els botons superiors de la pantalla central?

c) Instal·la els paquets "cockpit-file-sharing" i "nfs-utils". Inicia el servei `nfs-server` i reinicia sessió en el Cockpit. ¿Quina secció nova veus al menú lateral esquerra? ¿Què hi pots fer allà si hi accedeixes?

d) Vés a l'enllaç "View details & history" que apareix a la secció "Usage" del panell "Overview" i clica sobre el botó "Install cockpit-pcp". Un cop instal·lats els paquets necessaris, reinicia la sessió Cockpit i inicia el servei `pmlogger` ¿Què veus de diferent a la pàgina "Overview"? (hauràs d'esperar una estona per veure quelcom interessant)

dII) ¿Què explica aquest apartat de la documentació oficial de Cockpit: <https://cockpit-project.org/guide/latest/feature-pcp.html> ?

8.- Webmin (<http://webmin.com>) és una alternativa (més tradicional) a Cockpit. Vés a <https://doxfer.webmin.com/Webmin/Introduction> i https://doxfer.webmin.com/Webmin/Webmin_Modules i digues tres mòduls ("plugins") que tingui Webmin que no tingui Cockpit

NOTA: Una altra alternativa similar és <https://ajenti.org>