**Systemd**

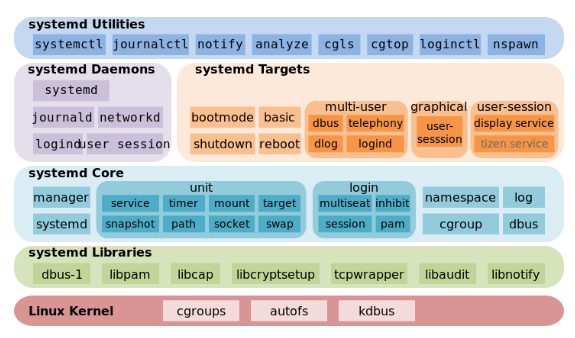
|  |  |  |  |
| --- | --- | --- | --- |
| **DESC** | | | |
| Gestionnaire général (gestion d’unités : services…, journalisation…) | | | |
| **FILES** | | | |
| /etc/ | inittab | | Obsolète avec systemd |
|  | Systemd/ | Journald.conf | Configuration des logs de journald |
|  |  | System/ | Lien symbolique vers la case en dessous  Modifier ici, ça prendra la priorité sur ci-dessous |
| /lib/Systemd/system/ | | | Unit files : init, stop…script  Ne pas modifier ces fichiers |
| /usr/lib/systemd/system/ | | |
| /run/systemd/system/ | | | Fichiers Unit crées dynamiquement.  Perdu lors du reboot |
| **QUICK START** | | | |
|  | | | |

**Généralités :**

* Successeur d’Init et upstart.
* Gestionnaire de service(boot…), de montage, d’états du système…
* Composants de systemd :
  + Journald : collecte et gère les logs du kernel + applications
* Targets : anciennement run-level, Etat qu’un service essaye d’atteindre.

**Override de certaines directives d’un fichier unit :**

* Systemctl edit <unité>
* Ajout d’un dossier programme.service.d juste à côté du fichier unit, puis X.conf dans ce dossier



**Types d’unités :**

|  |  |
| --- | --- |
| **Unit** |  |
| .service | Cette unité décrit comment gérer le service ou l’application :  Start, stop, conditions de démarrage, dépendances, infos… |
| .socket | Décrit un réseau, un IPC ou une FIFO et est associé à un service.  Systemd l’utilise pour l’activation d’un service par le réseau. |
| .device | Ce device doit être géré par systemd |
| .mount | Ce point de montage doit être géré par systemd |
| .automount | Déinit un point de montage automatiquement monté  Doit être associé à un .mount |
| .swap |  |
| .target | Points de synchronisation lors du boot ou d’un changement d’état du serveur (run-level).  Les autres unités spécifient leur relation à une target |
| .path | Systemd l’utilise pour l’activation d’un service par un chemin |
| .timer | Similaire à cron :  Un service peut être démarré quand le timer est atteint |
| .snapshot | Point de restauration |
| .slice | Associé à un cgroup : contrôle des ressources par services. |
| .scope | Utilisé pour gérer des sets de processus système |

**Syntaxe générale d’un fichier unit :**

#Case sensitive

[Section]

Directive1=value

Directive2=value

Directive3= #Reset de la valeur

…

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Directive** | | **Comments** |
| [Unit] |  | | Métadatas  Relation avec les autres unités  Première Section d’un fichier Unit |
|  | Description= | |  |
|  | Documentation= | | URLs de docs |
|  | Requires= | | Liste d’unités qui doivent être démarrés avec succès et dont cette unité dépend.  Si ces dépendances ne sont pas satisfaites, l’unité ne démarrera pas. |
|  | Wants= | | Similaire à Requires mais en moins strict : si les dépendances ne sont pas satisfaites, l’unité continuera de fonctionner quand même |
|  | Bindsto= | | Similaire à Requires, mais cette unité se terminera si les unités associés se termineront. |
|  | Before= | | Les unités ici ne seront pas démarrées avant l’unité courante |
|  | After= | | Les unités ici seront démarrées avant l’unité courante |
|  | Conflicts= | | Les unités qui ne peuvent pas fonctionnées en même temps |
|  | Condition= | | Autre conditions pour que l’unité démarre et fonctionne |
|  | Assert= | | Idem Condition mais plus strict |
| [Install] |  | | Dernière section d’un fichier Unit  Optionnelle : Définit le comportement de l’unité si elle est enabled ou disabled (lors du boot donc)  Les directives listées listent quoi faire si l’unité est enabled |
|  | WantedBy= | |  |
|  | RequiredBy= | |  |
|  | Alias= | |  |
|  | Also= | |  |
|  | DefaultInstance= | |  |
| [Service] |  | | Propre à l’unité service |
|  | Type= | simple |  |
|  |  | forking |  |
|  |  | oneshot |  |
|  |  | dbus |  |
|  |  | notify |  |
|  |  | idle |  |
|  | RemainAfterExit= | |  |
|  | PIDFile= | |  |
|  | BusName= | |  |
|  | NotifyAccess= | |  |
|  | ExecStart= | |  |
|  | ExecStartPre= | |  |
|  | ExecStartPost= | |  |
|  | ExecReload= | |  |
|  | ExecStop= | |  |
|  | ExecStopPost= | |  |
|  | RestartSec= | |  |
|  | Restart= | |  |
|  | TimeoutSec= | |  |
| [Socket] |  | |  |
|  | ListenStream= | |  |
|  | ListenDatagram= | |  |
|  | ListenSequentialPacket= | |  |
|  | ListenFIFO= | |  |
|  | Accept= | |  |
|  | SocketUser= | |  |
|  | SocketGroup= | |  |
|  | SocketMode= | |  |
|  | Service= | |  |
| [Mount] |  | |  |
|  | What= | |  |
|  | Where= | |  |
|  | Type= | |  |
|  | Options= | |  |
|  | SloppyOptions= | |  |
|  | DirectoryMode= | |  |
|  | TimeoutSec= | |  |
| [Automount] |  | |  |
|  | Where= | |  |
|  | DirectoryMode= | |  |
| [Swap] |  | |  |
|  | What= | |  |
|  | Priority= | |  |
|  | Options= | |  |
|  | TimeoutSec= | |  |
| [Path] |  | |  |
|  | PathExists= | |  |
|  | PathExistsGlob= | |  |
|  | PathChanged= | |  |
|  | PathModified= | |  |
|  | DirectoryNotEmpty= | |  |
|  | Unit= | |  |
|  | MakeDirectory= | |  |
|  | DirectoryMode= | |  |
| [Timer] |  | |  |
|  | OnActiveSec= | |  |
|  | OnBootSec= | |  |
|  | OnStartupSec= | |  |
|  | OnUnitActiveSec= | |  |
|  | OnUnitInactiveSec= | |  |
|  | OnCalendar= | |  |
|  | AccuracySec= | |  |
|  | Unit= | |  |
|  | Persistent= | |  |
|  | WakeSystem= | |  |
| [Slice] |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

Garbage :

Machin.service