×

Un server in casa con NixOS

Facciamoci un cloud personale, privato e facilmente riproducibile

Linux Day Napoli 2023 – 28 ottobre 2023

Che cos'è un home server?

Un **server** è un qualsiasi computer che si presta a **offrire servizi** ad altri computer sulla rete locale o su internet. Il server non è necessariamente il tipo di macchina, ma **il ruolo che svolge**.

Un **home server** è un server da casa per uso personale.

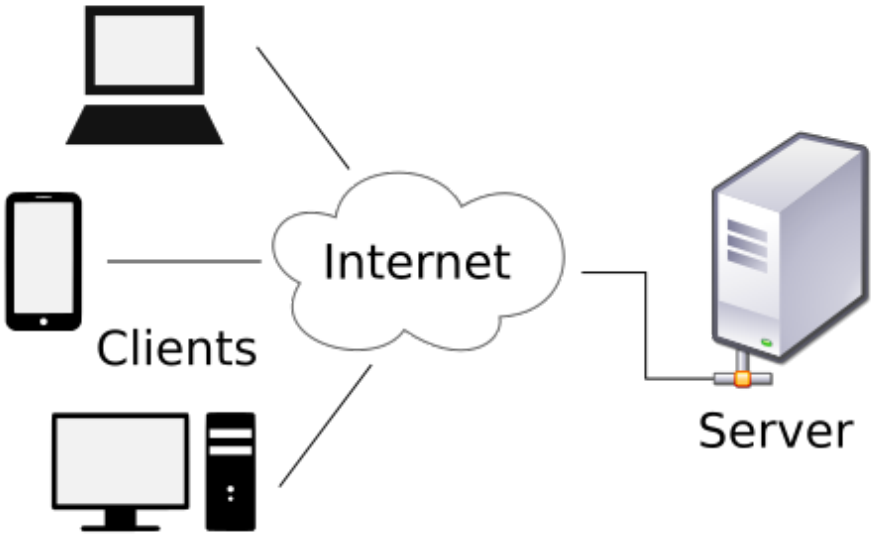


Immagine di utente Wikipedia **Calimo** (LGPL 2.1+)

Perché avere un home server?

Solitamente le persone usano **servizi online proprietari e centralizzati** per svolgere diverse attività (sincronia e condivisione di file e foto, streaming, chat e videochiamate, email, social network, collaborazione per lavoro, appunti e note, videoscrittura, ecc.) e questi servizi hanno molti problemi. Gran parte di questi problemi non ci sono su un home server.

Loghi appartenenti alle rispettive aziende

I vantaggi di un home server

- Flessibilità e scelta
- Controllo di hardware, software e dati
- Sicurezza e isolamento
- Costi per lo spazio di archiviazione
- Scalabilità e migrazione
- Divertimento (non per tutti)

Immagine promozionale di un HP MicroServer N36L

Come si fa un home server?

I tre punti di cui preoccuparsi:

- Hardware
- Sistema operativo e applicazioni
- Configurazione di rete (dominio, port forwarding, ecc.)

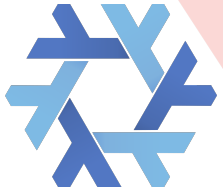
Hardware

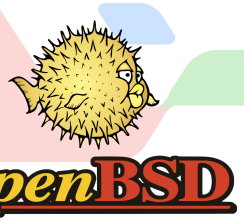
Come hardware può bastare **un qualsiasi computer**, da un vecchio desktop o portatile ad un single board computer (es. Raspberry Pi), che può stare acceso e connesso ad internet (preferibilmente con cavo ethernet) **24/7**.

Non c'è bisogno di hardware potentissimo a meno che il server non debba fare attività realmente pesanti. Per un home server solitamente non c'è bisogno di molta potenza.

Foto di un ThinkPad X220 e di un Raspberry Pi 3B

Sistema operativo

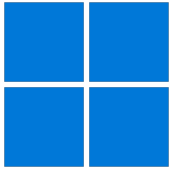




Linux è **il re indiscusso dei server**, permette di avere un ambiente minimale e pulito con una grande quantità di software utili ai server. Fra le distribuzioni più usate in questo ambito abbiamo **Ubuntu**, **Debian**, **RHEL** e **CentOS**, noi però useremo qualcosa di diverso e inusuale…

È comunque possibile usare un altro sistema per uso da home server, come un ***BSD** o persino **Windows** e **macOS**. Questi ultimi due non hanno tutto il supporto software da server che gli altri hanno, pertanto **potrebbero essere limitati in funzionalità**, perché sono sistemi desktop-first.

Il classico processo per installare un sistema operativo è scaricare la ISO, flasharla su una chiavetta USB, avviare il server da questa chiavetta e procedere con l'installer fornito dal sistema.



Loghi appartenenti alle rispettive organizzazioni

Configurazione di rete

- Ottenere un dominio (registrar o DNS dinamico)
- Usare un servizio di DNS dinamico (se il tuo IP pubblico è dinamico)
- Impostare un IP locale statico per il server
- Aprire le porte del router (port forwarding)

namecheap

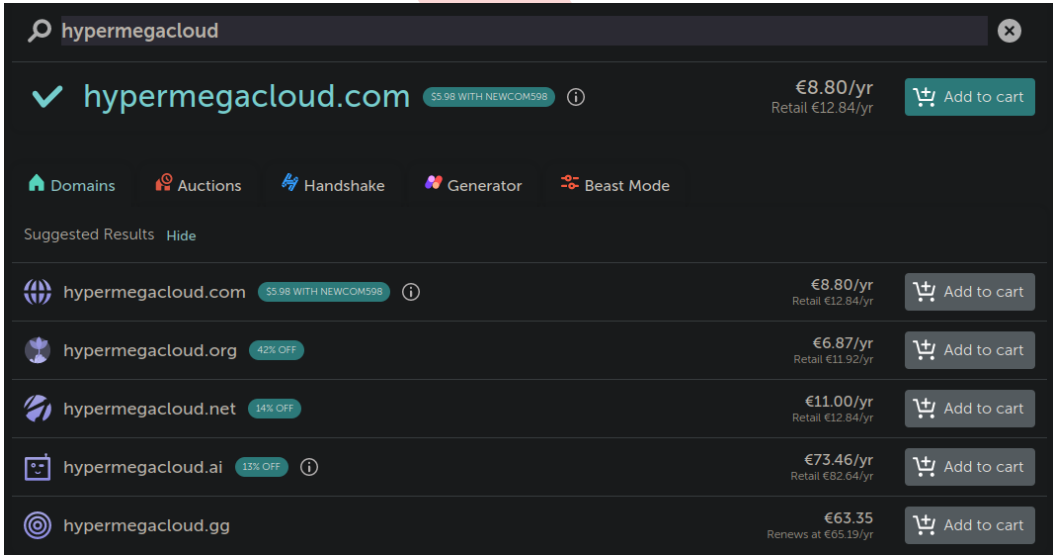
FreeDNS

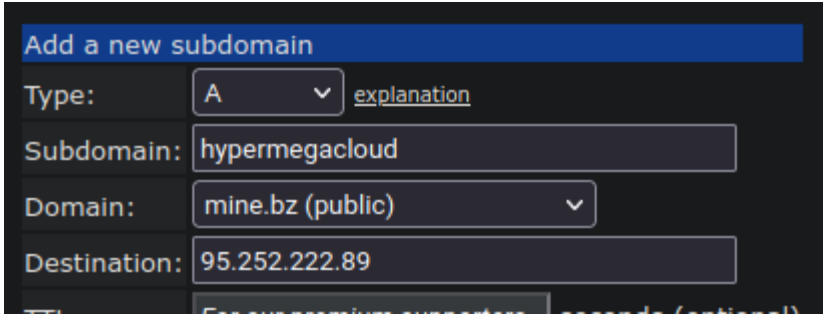
Loghi appartenenti alle rispettive aziende; foto di un router TIM HUB+

Ottenere un dominio

Un dominio è una stringa associata ad un indirizzo IP. Solitamente i domini si acquistano dai **registrar** per una certa cifra ogni anno (es. *`hypermegacloud.com`*).

In alternativa si può usare un servizio di DNS dinamico per ottenere un **sottodominio gratuito** con loro (es. *`hypermegacloud.mine.bz`*).



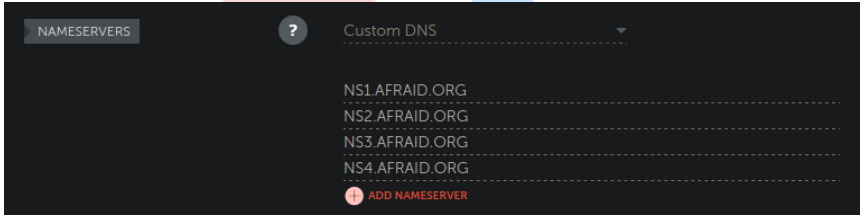


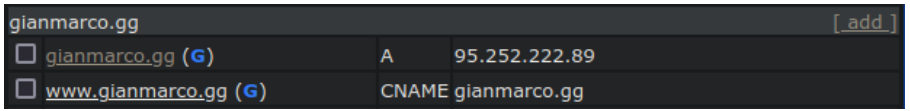
DNS dinamico

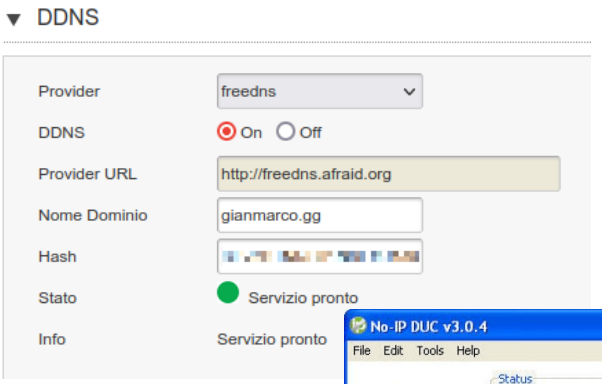
Un servizio di DNS dinamico aiuta ad **associare il tuo dominio** (o sottodominio) **al tuo IP pubblico**. Nel caso di un dominio, bisogna andare nel pannello di controllo del proprio registrar e impostare i **nameserver** del dominio come i nameserver del servizio.

Nel pannello di controllo del servizio di DNS dinamico bisogna aggiungere il dominio e creare almeno un “**A” record** che punta il dominio al proprio IP pubblico. È meglio anche creare un **“CNAME” record** che punta il sottodominio “www” al primo record.

La maggior parte delle persone ha un **IP pubblico dinamico**, ossia che cambia ogni *X* giorni o mesi, quindi è meglio **automatizzare l’associazione** per gli IP nuovi. Solitamente si fa con un *cronjob* (uno script automatico con un tempo determinato), un’applicazione dedicata (come quella di No-IP) o una funzione speciale del proprio router.







IP locale statico

L'IP locale è un indirizzo associato a ogni dispositivo connesso alla rete locale **per poter essere trovati** (solitamente nel formato *192.168.1.x*). Anche il proprio home server ne ha uno, ma questo può cambiare automaticamente, perciò impostarne uno **statico** lo renderà **sempre uguale** (es. *192.168.1.12*).

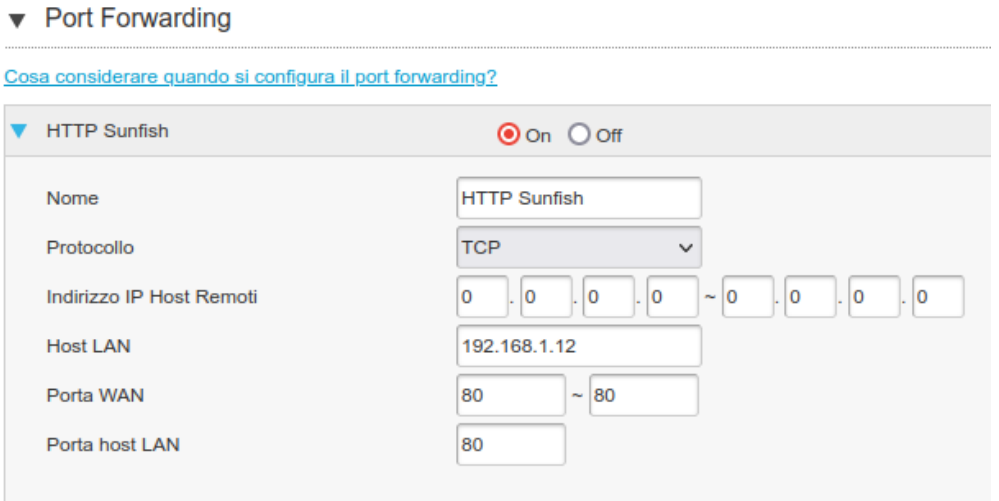
Questo può essere fatto tranquillamente nel pannello di controllo del router, accessibile dal browser solitamente all'IP *192.168.1.1*, e andando nella sezione apposita (può variare a seconda di che router hai). In alternativa l'IP locale statico si può configurare nel sistema operativo dell'home server.

Port forwarding

Port forwarding significa **aprire le porte del router** in modo da poterci accedere dall'esterno. Questo si fa nel pannello di controllo del router andando sulla sezione apposita (può variare a seconda di che router hai).

Le principali porte da aprire sono la **80** (HTTP), **443** (HTTPS) e **22** (SSH, non usata su Windows), associate ovviamente all'IP locale del proprio home server. Si possono aggiungere altre porte in base ai servizi che si installano sul server.

È bene evitare di lasciare aperte all'esterno porte inutilizzate, poiché può essere rischioso rimanere vulnerabili a causa di servizi che non ci servono e che potrebbero avere **falle di sicurezza**. Questo vale anche per il firewall del sistema operativo.



Che cos'è NixOS?

NixOS è una distribuzione Linux molto particolare. È incentrata su **Nix**, il suo **package manager**, ed è configurabile tramite **Nix**, il suo **linguaggio dichiarativo**.

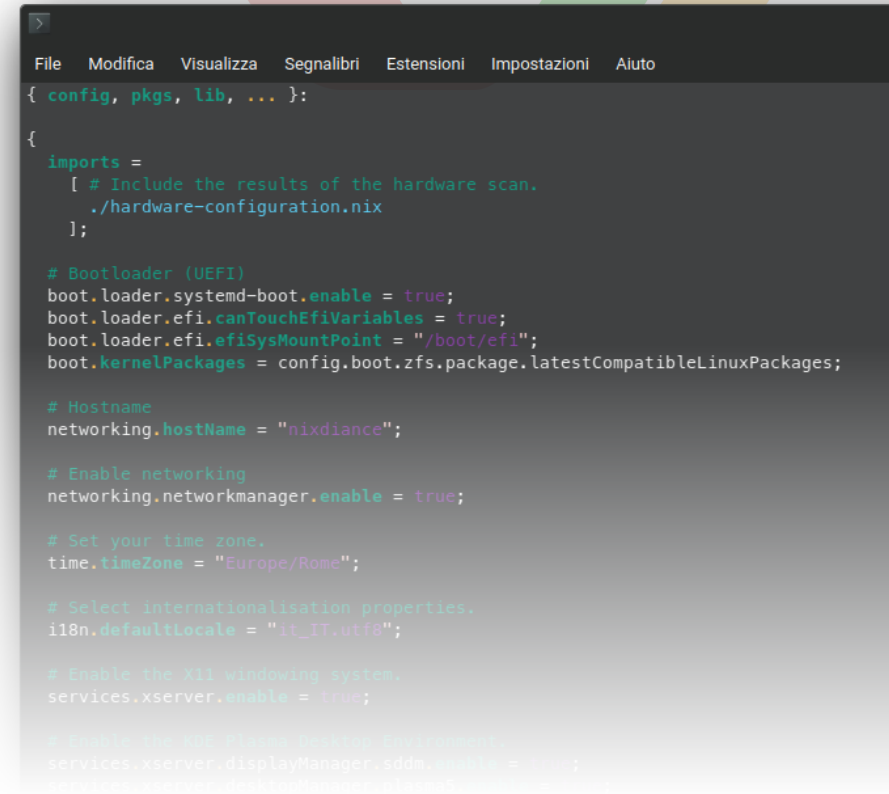
Lo sviluppo iniziò per mano di Eelco Dolstra nel 2003 come progetto di ricerca per poter ottenere un modello di sviluppo software **riproducibile** e **senza *dependency hell***.

Negli ultimi tempi, NixOS ha cominciato a diventare una distro molto popolare, specialmente fra programmatori e smanettoni che hanno più di 2 o 3 computer che hanno **configurazioni identiche**.

Differenze da altre distro

NixOS è diversa dalle altre distribuzioni:

- Configurazione dichiarativa
- Configurazione riproducibile
- Aggiornamenti atomici e rollback
- Sistema immutabile
- Pacchetti multi-utente e testabili al volo
- Non conforme al Filesystem Hierarchy Standard



Esempio di configurazione

Alcuni esempi basilari di come configurare NixOS per un home server:

- Configurazioni di base
- Disco condiviso in rete (Samba)
- Web server (nginx)
- Altre applicazioni possibili
- Servizi di systemd personalizzati

Tutte queste configurazioni si fanno nel file `/etc/nixos/configuration.nix`.

Configurazioni di base

```
# Modifica questo file di configurazione per definire cosa va installato sul
# tuo sistema.  L'aiuto è disponibile nella pagina man configuration.nix(5)
# e nel manuale di NixOS (accessibile facendo `nixos-help`).

{ config, pkgs, ... }:

{
  imports =
    [ # Includi i risultati della ricerca hardware.
      ./hardware-configuration.nix
    ];
```


Configurazioni di base

```
# Usa il boot loader EFI systemd-boot.
boot.loader.systemd-boot.enable = true;
boot.loader.efi.canTouchEfiVariables = true;

networking.hostName = "serverino"; # Definisci il tuo hostname.
# Scegli solo una delle due opzioni per la rete.
# networking.wireless.enable = true;  # Abilità le reti senza fili con wpa_supplicant.
networking.networkmanager.enable = true;  # Più facile da usare e su molte distro è il default.

# Imposta il tuo fuso orario.
time.timeZone = "Europe/Rome";
```


Configurazioni di base

```
# Configura un proxy di rete se necessario
# networking.proxy.default = "http://user:password@proxy:port/";
# networking.proxy.noProxy = "127.0.0.1,localhost,internal.domain";


# Seleziona le proprietà di internazionalizzazione.
i18n.defaultLocale = "it_IT.UTF-8";
console = {
  font = "Lat2-Terminus16";
  keyMap = "it";
  useXkbConfig = true; # usa xkbOptions nella tty.
};
```


Configurazioni di base

```
# Abilita il sistema di finestre X11.
services.xserver.enable = true;
services.xserver.desktopManager.gnome.enable = true;

# Configura la tastiera su X11.
services.xserver.layout = "it";
# services.xserver.xkbOptions = "eurosign:e,caps:escape";

# Abilita CUPS per stampare documenti.
# services.printing.enable = true;
```


Configurazioni di base

```
  # Abilita l'audio.

  sound.enable = true;

  hardware.pulseaudio.enable = true;


  # Abilita il supporto al touchpad (abilitato per default su gran parte dei
desktopManager).

  # services.xserver.libinput.enable = true;
```


Configurazioni di base

```
# Definisci un account utente. Non dimenticarti di impostare una password con ‘passwd’.
users.users.gianmarco = {
  isNormalUser = true;
  extraGroups = [ "wheel" ]; # Abilita ‘sudo’ per l’utente.
  packages = with pkgs; [
    firefox
    tree
  ];
  passwordFile = "/locazione/segreta/del/passwordfile";
};
```


Configurazioni di base

```
  # Lista dei pacchetti installati nel profilo di sistema. Per cercare, fai:
  # $ nix search wget
  environment.systemPackages = with pkgs; [
    vim # Non dimenticare di aggiungere un editor per modificare configuration.nix! L'editor Nano è anche installato per default.
    wget
    screen
    xonotic-dedicated
  ];
```


Configurazioni di base

```
# Lista dei servizi che vuoi abilitare:


# Abilita il demone OpenSSH.
services.openssh.enable = true;


# Apri le porte nel firewall.
networking.firewall.allowedTCPPorts = [ 80 443 22 ];
networking.firewall.allowedUDPPorts = [ 80 443 22 ];
# O disabilita il firewall completamente.
# networking.firewall.enable = false;
```


Configurazioni di base

```
  # Copia il file di configurazione di NixOS e collegalo dal sistema
ottenuto
  # (/run/current-system/configuration.nix). Questo è utile nel caso tu
  # cancellassi configuration.nix accidentalmente.
  system.copySystemConfiguration = true;
```


Configurazioni di base

```
  # Questo valore determina la versione di NixOS dalla quale le impostazioni
  # di default per i dati degli stati, come percorsi di file e versioni dei database
  # sul tuo sistema sono state prese. È perfettamente OK e raccomandato di lasciare
  # questo valore alla versione della prima installazione di questo sistema.
  # Prima di cambiare questo valore leggi la documentazione per questa opzione
  # (es. man configuration.nix oppure su https://nixos.org/nixos/options.html).
  system.stateVersion = "23.05"; # Hai letto il commento?


}
```


Disco condiviso in rete (Samba)

```
# File system.
fileSystems."/mnt/discone" = {
  device = "/dev/sdb1";
  fsType = "ext4";
  options = [ "defaults" "user" "rw" "nofail" ];
};
```


Disco condiviso in rete (Samba)

```
# Samba.
services.samba = {
  enable = true;
  securityType = "user";
  openFirewall = true;
  extraConfig = ''
    workgroup = WORKGROUP
    server string = serverino
    netbios name = serverino
    smb encrypt = required
    security = user
  '';
```


Disco condiviso in rete (Samba)

```
shares = {
  discone = {
    path = "/mnt/discone";
    browseable = "yes";
    "read only" = "no";
    "guest ok" = "yes";
  };
```

Disco condiviso in rete (Samba)

```
    home = {
      path = "/home/gianmarco";
      browseable = "yes";
      "read only" = "no";
      "guest ok" = "no";
      "force user" = "gianmarco";
    };
  };
};
```

Web server (nginx)

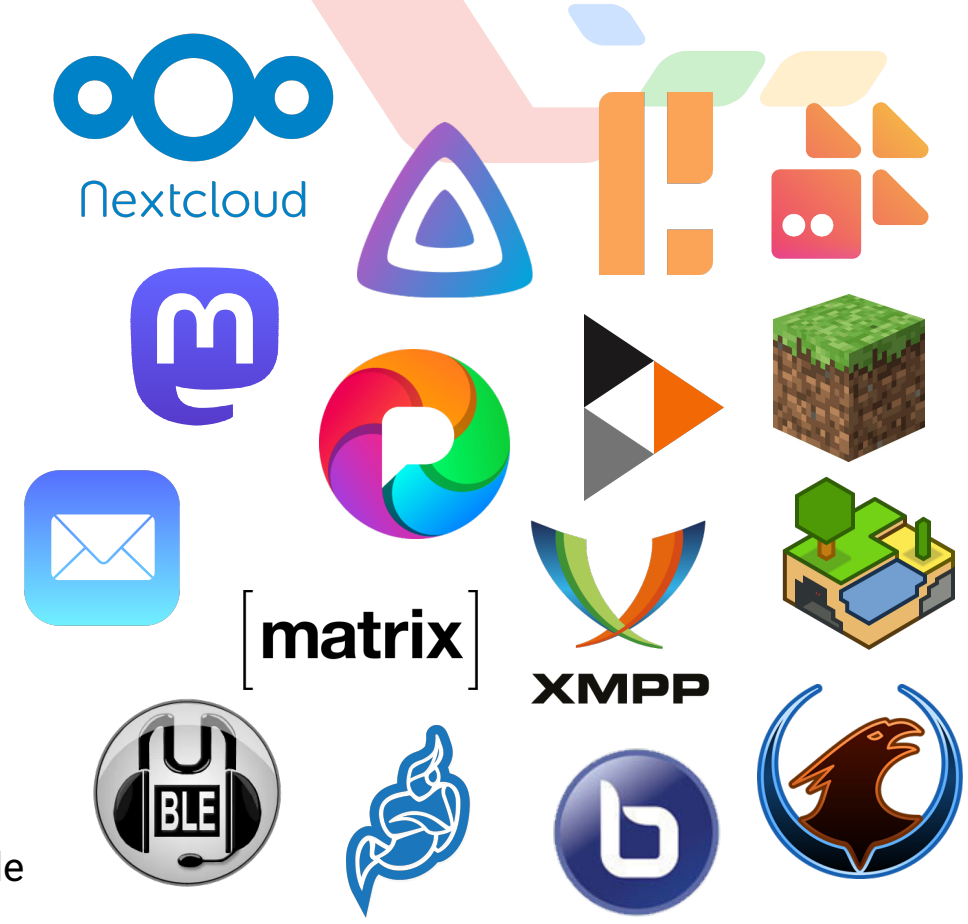
```
# Nginx.
services.nginx.enable = true;
services.nginx.virtualHosts = {
  "hypermegacloud.com" = {
    addSSL = true;
    enableACME = true;
    forceSSL = true;
    root = "/srv/nginx/hypermegacloud.com";
  };
};
```


Altre applicazioni possibili

Si possono installare e configurare molti servizi:

- Nextcloud come cloud personale
- Jellyfin per lo streaming di film, serie e musica
- Social del fediverso (Mastodon, Peertube, ecc.)
- Server di posta elettronica
- Server di chat (Matrix, XMPP, ecc.)
- Server di gioco (Xonotic, Minecraft, Minetest, ecc.)
- Server di chiamate (Mumble, Jitsi, ecc.)

Ovviamente tutto ciò si configura dichiarativamente nel file di configurazione.



Loghi appartenenti alle rispettive organizzazioni

Servizi di systemd personalizzati

```
# Server Xonotic.
systemd.services.xonoticServer = {
  wantedBy = [ "multi-user.target" ];
  after = [ "network.target" ];
  description = "Avvia il server Xonotic in una sessione di GNU Screen.";
  serviceConfig = {
    Type = "forking";
    User = "gianmarco";
    ExecStart = ''${pkgs.screen}/bin/screen -dmS xon ${pkgs.xonotic-dedicated}/bin/xonotic-dedicated'';
    ExecStop = ''${pkgs.screen}/bin/screen -S xon -X quit'';
  };
};
```


Comandi importanti

Alcuni comandi importanti per la gestione del sistema:

- Ricompilare la configurazione
- Aggiornare tutti i pacchetti
- Upgrade ad una nuova versione di NixOS
- Generazioni di sistema e rollback
- Cercare pacchetti
- Gestire pacchetti a mano nell’ambiente utente
- Provare pacchetti in una shell isolata

Ricompilare la configurazione

Ogni volta che si modifica il file di configurazione di sistema, bisogna dire a NixOS di ricompilare la configurazione per applicare le modifiche fatte. Questo si fa con il comando:

```
sudo nixos-rebuild switch
```

Solitamente la ricompilazione dura pochi secondi e raramente richiede di riavviare il sistema, ciò permette di avere **meno downtime possibile**.

Aggiornare tutti i pacchetti

Per aggiornare tutti i pacchetti, basta dare i comandi:

```
sudo nix-channel --update
```

```
sudo nixos-rebuild switch
```

per ricaricare il database locale e ricompilare il sistema in base alle nuove versioni dei pacchetti che saranno scaricate e installate durante la ricompilazione.

Upgrade ad una nuova versione di NixOS

Viene rilasciata una nuova versione di NixOS **ogni 6 mesi***, similmente ad Ubuntu, che porta aggiornamenti maggiori ai pacchetti. Fare l'upgrade è un processo leggermente più complicato:

```
sudo nix-channel --remove nixos

sudo nix-channel --add https://nixos.org/channels/nixos-23.05 nixos

sudo nix-channel --update

sudo nixos-rebuild switch
```

Così il sistema sarà aggiornato alla versione specificata dopo aver riavviato.

* esiste anche il canale **`unstable`** che è rolling, similmente ad Arch.

Upgrade ad una nuova versione di NixOS

Attenzione: ogni volta che si fa un upgrade ad una nuova versione, **potrebbero essere necessari cambiamenti al file di configurazione** poiché potrebbero esserci cambiamenti di sintassi, pacchetti rinominati, opzioni rimosse, ecc., quindi occhio ai changelog (https://nixos.org/blog/announcements.)

Generazioni di sistema e rollback

NixOS salva una snapshot di ogni stato del sistema ad ogni ricompilazione, queste snapshot sono dette **generazioni**, per averne la lista si dà il comando:

```
sudo nix-env --list-generations --profile /nix/var/nix/profiles/system
```

dove si specifica il profilo di un utente o, come in questo caso, il profilo di sistema. I profili sono **liste di generazioni** con collegamenti simbolici alle configurazioni e ai pacchetti precedenti.

Per fare un rollback alla generazione di sistema precedente si dà il comando:

```
sudo nixos-rebuild --rollback switch
```

Questo è comodo dopo una ricompilazione o un aggiornamento che dà problemi.

Cercare pacchetti

Per cercare pacchetti dalla riga di comando si dà il comando:

`nix search nixpkgs *pacchetto*`

dove *`pacchetto`* è il nome del pacchetto che si vuole trovare.

Gestire pacchetti a mano nell'ambiente utente

È possibile adoperare pacchetti sotto il proprio utente piuttosto che sul sistema.

Per installare un pacchetto si dà: **`nix-env -iA pacchetto`**

Per listare i pacchetti installati si dà: **`nix-env -q`**

Per disinstallare un pacchetto si dà: **`nix-env -e pacchetto`**

Per aggiornare tutti i pacchetti si dà: **`nix-env -u`**

Questi pacchetti **non saranno presi in considerazione dalle ricompilazioni** di sistema, nemmeno dalla lista dei pacchetti utente nel file di configurazione.

Provare pacchetti in una shell isolata

È possibile provare un pacchetto senza installarlo sul sistema o nell'ambiente utente, quindi installandolo temporaneamente su una shell isolata. Questo si fa con il comando:

`nix-shell -p` *`pacchetto`*

Questo aprirà una nuova shell dove si potrà usare il pacchetto finché non viene chiusa la shell.

Altre risorse utili

Scoprire NixOS può essere una vera odissea che non si può affrontare tutta in una volta, ma fortunatamente ci sono molte risorse utili online:

- Il sito ufficiale di Nix e NixOS (https://nixos.org/learn)
- Il manuale di Nix e NixOS (https://nixos.org/manual/nixos/stable oppure **`nixos-help`**)
- La wiki community (https://nixos.wiki)
- La ricerca di pacchetti e opzioni (https://search.nixos.org)
- Home Manager (https://github.com/nix-community/home-manager)
- Nix Flakes (https://nix.dev/concepts/flakes)

Altri usi per NixOS

NixOS è utile anche al di fuori dell'ambito server o home server, molta gente lo usa per i **propri computer personali o da lavoro**, per **dispositivi embedded** anche basati su ARM, per avere un **ambiente di sviluppo software** coerente ed estensibile, si può usare praticamente per qualsiasi cosa. Forse un giorno lo vedremo anche sui telefoni con Linux (es. Pinephone).

Screenshot di utente Reddit **u/horriblesmell420** da r/unixporn; foto di un Pinephone Pro con Plasma Mobile

Dimostrazione di Artemis

Artemis è **il mio nuovo home server**. Ho riutilizzato un vecchio desktop con un Pentium di generazione Skylake e 16 GB di RAM DDR4 per farmi un home server più potente e, ovviamente, riproducibile e facile da mantenere grazie a **NixOS**.

Gli ho comprato **un nuovo case compatto** (iTek Evoke / SAMA 01) che mi permette di tenerlo su uno scaffale e di poter aggiungere degli hard disk in futuro.

Immagine promozionale del case iTek EVOKE

Artemis sullo scaffale

Artemis a case aperto

La mia shell


Last login: Sat Aug 5 16:01:34 2023 from 176.200.238.29
gianmarco@artemis
os NixOS 23.05 (Stoat)
host x86_64
kernel 6.3.12
uptime 12d 17h 5m
pkgs 708
memory 1546M / 15872M
neofetch
gianmarco@artemis
OS: NixOS 23.05.2517.e9ca92b55be (Stoat) x86_64
Host: ASUSTeK COMPUTER INC. H110M-A
Kernel: 6.3.12
Uptime: 12 days, 17 hours, 5 mins
Packages: 708 (nix-system)
Shell: zsh 5.9
Terminal: /dev/pts/1
CPU: Intel Pentium G4400 (2) @ 3.300GHz
GPU: Intel HD Graphics 510
Memory: 1882MiB / 15872MiB

La configurazione di sistema

artemisconfig.nix

home > gianmarco > Scrivania > artemisconfig.nix

```nix
# Artemis NixOS configuration.

{ config, pkgs, lib, ... }:

{
  imports =
    [ # Include the results of the hardware scan.
      ./hardware-configuration.nix
    ];

  # Use the systemd-boot EFI boot loader and use the latest kernel compatible with OpenZFS.
  boot.loader.systemd-boot.enable = true;
  boot.loader.efi.canTouchEfiVariables = true;
  boot.kernelPackages = config.boot.zfs.package.latestCompatibleLinuxPackages;

  # Hostname.
  networking.hostName = "artemis";
  # Network service.
  networking.networkmanager.enable = true;

  # Time zone.
  time.timeZone = "Europe/Rome";

  # Select internationalisation properties.
  i18n.defaultLocale = "it_IT.UTF-8";
  console = {
    font = "Lat2-Terminus16";
    keyMap = "it";
    #useXkbConfig = true; # use xkbOptions in tty.
  };
```

Oltre 500 righe!!!

tput Cerca Progetto Terminale Diagnostica 1:1 INSERISCI Tabulazione con spazi: 2 (4) UTF-8 Nix

Cosa ci gira attualmente

- Il mio sito principale (https://gianmarco.gg) ed altre sezioni
- Il mio server Git (https://git.gianmarco.gg)
- Un nodo Syncthing
- Un server di gioco per Xonotic e uno per Minecraft
- Il mio server XMPP, Mumble e Gotify
- Il mio server Peertube (https://videos.gianmarco.gg)
- Il mio server Owncast (https://live.gianmarco.gg)
- Una capsula Gemini (gemini://gianmarco.gg)

Piani futuri

- Migrare tutti gli altri servizi (gli altri siti, Nextcloud, Matrix, Jellyfin, Isso, SearXNG, ecc.)
- Aggiungere qualche hard disk da 16 o 18 TB con OpenZFS
- Sviluppare una strategia di ridondanza (RAID o Snapraid + MergerFS)
- Sviluppare una strategia di backup anche per altri dispositivi
- Impostare un nodo Monero privato
- Impostare un client di torrent headless (AGCOM permettendo)

www.gianmarco.gg
www.nalug.tech