آکادمی کندوی دانش
https://learninghive.ir

راهنمای جامع دسترسی و مدیریت منابع نرم‌افزاری در لینوکس

در شرایط خاص

به همراه تنظیمات در ویندوز



- مخازن لینوکس (APT, DNF, Pacman) و میرورهای داخلی
- DNS و تنظیم آن در سیستم‌عامل‌ها
- پراکسی و دسترسی پایدار به اینترنت
- مدیریت بسته‌ها و ابزارهای توسعه
- همگام‌سازی زمان با سرورهای داخلی
- راهکارهای دسترسی در شرایط محدودیت

نویسنده:
حسین سیلانی

در شرایط خاص

به همراه تنظیمات در ویندوز

مرور سریع کتاب

این کتابچه یک «جعبه‌ابزار بقا» برای توسعه‌دهندگان و مدیران سیستم است ، این سند به شما یاد می‌دهد چگونه:

- **مخازن لینوکس (APT, DNF , Pacman)** را به سرورهای داخلی (مثلIUT ، آروان، کرگدن، لیارا و غیره تغییر دهید تا بتوانید نرم‌افزار نصب کنید.
- **DNS**سیستم را به سرورهای داخلی تغییر دهید تا سایت‌های ایرانی باز شوند.
- **پکیج‌منیجرهای زبان‌های برنامه‌نویسی** (pip, npm, Maven, Dockerو...را به میرورهای ایرانی متصل کنید تا پروژه‌های برنامه‌نویسی متوقف نشوند.
- از **سرویس‌های آپلود داخلی** برای به اشتراک‌گذاری فایل استفاده کنید.
- در صورت نیاز، **پروکسی (Proxy)** را برای ابزارهای مختلف تنظیم کنید.
- ساعت سیستم **(NTP)** را با سرورهای زمانی داخلی همگام کنید تا گواهی‌های امنیتی (SSL) و لاگ‌ها دچار خطا نشوند.

مقدمه و سلب مسئولیت

این کتابچه (راهنمای جامع دسترسی و مدیریت منابع نرم‌افزاری در لینوکس) بنا بر شرایط خاص داخل تهیه و تدوین شده است.هدف اصلی این مجموعه، کمک به دانشجویان، پژوهشگران و توسعه‌دهندگان است تا در این شرایط سخت، بتوانند کارهای آموزشی و پروژه‌های داخلی خود را پیش ببرند. تمام مطالب و آدرس‌های این کتاب فقط برای اهداف آموزشی و استفاده در پروژه‌های داخلی جمع‌آوری شده است.

سلب مسئولیت:

نویسنده این کتاب هیچ مسئولیتی در قبال نحوه استفاده از این اطلاعات ندارد. هرگونه دسترسی غیرمجاز یا استفاده نادرست از مطالب این کتاب، کاملاً به عهده خود کاربر است و نویسنده در قبال آن پاسخگو نیست.

حسین سیلانی

۱۴۰۵

راه های ارتباطی

🌐 وبسایت شخصی، رزومه و پروژه‌ها

https://seilani.ir/

🐝 آکادمی کندوی دانش | آموزش لینوکس و متن‌باز

https://learninghive.ir/

🔷 کانال در اپلیکیشن بله:

http://ble.ir/linuxtnt

🔷 کانال تلگرام:

https://t.me/linuxtnt

فهرست

۱ .....

راهنمای جامع دسترسی و مدیریت منابع نرم‌افزاری در لینوکس ۲

در شرایط خاص ۲

به همراه تنظیمات در ویندوز ۲

مقدمه و سلب مسئولیت ۳

بخش اول ۱۲

لیست فایل‌های ISO قابل دسترس در سرورهای داخلی ایران ۱۳

بخش دوم ۱۵

لیست dns های جایگزین ۱۶

DNSهای پیشنهادی اپراتورهای اینترنت ۱۶

۳.DNSهای مرتبط با مؤسسات دانشگاهی و پژوهشی ایران ۱۸

۴.جدول DNS های پیشنهادی (از داخل ایران) ۱۹

بخش سوم ۲۰

روش تظیم dns در ویندوز ۲۱

روش تنظیم DNS در لینوکس ۲۲

در ابتدا روش های خالی کردن کش dns ۲۲

تشخیص سرویس کشDNS ۲۲

#بررسی فایلresolv.conf ۲۲

دستورات پاک کردن کش بر اساس سرویس ۲۲

خالی کردن کش dns مرورگرها ۲۳

بخش چهارم ۲۴

تنظیمات dns جدید در لینوکس ۲۵

اوبونتو(Ubuntu 26.04/22.04/24.04) ۲۵

دبیان(Debian 11/12) ۲۶

تنظیم دائمی DNSدر دبیان با استفاده از NetworkManager ۲۷

تنظیم دائمی DNSدر دبیان با استفاده از systemd-resolved ۲۹

تنظیم DNS در آرچ لینوکس(Arch Linux) ۳۴

بخش پنجم ۳۸

جایگزینی مخازن و سرورهای داخلی برای توزیع‌های لینوکس ۳۹

فهرست جامع مخازن داخلی ایران ۳۹

Debian ۴۰
Ubuntu ۴۰
Arch Linux ۴۰
Fedora ۴۱
Rocky Linux ۴۱
openSUSE ۴۱
Alpine Linux ۴۱
Linux Mint ۴۲
Manjaro ۴۲
Kali Linux ۴۲
CentOS ۴۲
FreeBSD ۴۲
Raspbian ۴۲
بخش ششم ۴۳
مخازن داخلی برای توزیع Debian ۴۴
سایر مخازن جایگزین برای دبیان ۴۶
مخازن داخلی برای توزیع اوبونتو ۴۸
سایر مخازن داخلی برای Ubuntu 26.04 ..... ۴۹
مخازن داخلی برای توزیع آرچ لینوکس ۵۱
آدرس دقیق میرور هر ارائه‌دهنده ۵۱
سایرمخازن ۵۲
مخازن داخلی برای توزیع فدورا ۵۳
سایر مخازن برای فدورا ۵۳
مخازن داخلی برای توزیع راکی لینوکس ۵۵
سایرمخازن برای راکی لینوکس ۵۵
مخازن داخلی برای توزیع مینت ۵۷
مخزن Kargadan برای مینت ۵۷
مخازن داخلی برای توزیع Alpine Linux ۵۸
روش کلی تغییر مخزن ۵۸
سایر مخازن الپاین ۵۹
تکمیل سایر میرورهای Alpine ۵۹

.به‌روزرسانی مخازن ۵۹
مخازن داخلی برای توزیع Manjaro Linux ۶۰
مخازن داخلی برای توزیع – Kali Linux ۶۲
CentOS میرورهای ایرانی ۶۳
مخازن داخلی برای توزیعFreeBSD ۶۳
مخازن داخلی برای توزیع– Raspbian ۶۴
مخازن داخلی برای توزیعRed Hat Enterprise Linux (UBI) ۶۵
بخش هفتم ۶۸
جایگزینی مخازن بسته‌های نرم‌افزاری(Package Managers) ۶۹
Python (pip / PyPI) ۶۹
JavaScript / Node.js (npm) ۷۰
Yarn ۷۱
Gradle / Maven .اندروید و جاوا ۷۱
جمکو(Jamko) ۷۳
سایر بسته‌های نرم‌افزاریNuGet ، Composer، Go ۷۳
PHP / Composer ۷۵
Go (Go Modules) ۷۶
.NET / NuGet.۲،۸ ..... ۷۶
Docker ۷۷
راهنمای استفاده از میرور Kargadanبرای زبان‌های برنامه‌نویسی و ابزارهای توسعه ۷۸
۱. – Python استفاده از مخزن PyPI ۷۸
۲. – Conda استفاده از مخزن Conda-Forge ۷۸
۳. – Node.js استفاده از مخزن npm ۷۹
۴. – Yarn استفاده از مخزن Yarn ۷۹
۵. – Docker تنظیم Registry Mirror ۸۰
۶. – Go استفاده از Go Modules Proxy ۸۰
۷. – Maven استفاده از مخزن Maven Central ۸۱
۸. – .NET استفاده از مخزن NuGet ۸۱
۹. – PHP استفاده از مخزن Composer ۸۲
تنظیم میرورهای DockerHub ۸۳
سرویس میرور رجیستری گیت هاب ۸۳
سرویس میرور رجیستری Quay ۸۴

سرویس میرور رجیستری Kubernetes ۸۵
سرویس میرور PyPI لیارا ۸۶
سرویس میرور npm ۸۶
بخش هشتم ۸۷
سرویس‌های آپلود و اشتراک‌گذاری فایل ایرانی ۸۸
بخش نهم ۸۹
مدیریت Proxy و دسترسی شبکه در لینوکس ۹۰
۱. مقدمه ۹۰
۲. آشنایی با مفاهیمProxy ، Mirror، Cache وRepository ۹۰
۲,۱.Proxy ۹۰
۲,۲.Mirror ۹۰
۲,۳.Cache ۹۱
۲,۴.Repository ۹۱
۲,۵. مقایسه مفاهیم ۹۲
۳. انواعProxy ۹۲
۳,۱.HTTP Proxy ۹۲
۳,۲.HTTPS Proxy ۹۲
۳,۳.SOCKS Proxy ۹۳
۴. تنظیم Proxy در محیط لینوکس ۹۳
۴,۵. غیرفعال کردنProxy ۹۵
۴,۶. دائمی کردن تنظیماتProxy ۹۵
۵. تنظیم Proxy برای APT در Debian وUbuntu ۹۵
۵,۱. تنظیم Proxy با نام کاربری و رمز عبور ۹۶
۵,۲. غیرفعال کردن Proxy برای یک مخزن ۹۶
۶. تنظیم Proxy برای DNF در Fedora و توزیع‌های مبتنی برRPM ۹۷
۶,۱. تنظیم Proxy برای یک مخزن مشخص ۹۷
۷. تنظیم Proxy برای Pacman در Arch Linux وManjaro ۹۸
۸. تنظیم Proxy برایGit ۹۹
۸,۱. تنظیم Proxy فقط برای یک مخزن ۹۹
۸,۲. استفاده از SOCKS5 درGit ۹۹
۹. تنظیم Proxy برایPython وpip ۱۰۰
۹,۱. تنظیم Proxy هنگام نصب بسته ۱۰۰
۹,۲. تنظیم دائمیProxy ۱۰۰

۹٫۳. استفاده از Proxy همراه باMirror ۱۰۰
۱۰. تنظیم Proxy برایnpm ۱۰۰
۱۰٫۱. ترکیب Proxy وRegistry ۱۰۱
۱۱. تنظیم Proxy برایDocker ۱۰۱
۱۱٫۱. تنظیم Proxy برایDocker Daemon ۱۰۱
۱۱٫۲. تنظیم Proxy باSystemd ۱۰۲
۱۱٫۳. تنظیم Proxy برایContainer ها ۱۰۲
۱۲. استفاده از SOCKS5 در لینوکس ۱۰۳
۱۲٫۱. تنظیم SOCKS5 برای نشست ترمینال ۱۰۳
۱۲٫۲. آزمایش ارتباط باSOCKS5 ..... ۱۰۳
۱۲٫۳. تفاوت socks5 وsocks5h ۱۰۳
۱۲٫۴. محدودیت استفاده ازSOCKS5 ..... ۱۰۳
۱۳. بررسی محدودیت‌های DNS وTLS ۱۰۴
۱۳٫۱. بررسیDNS ۱۰۴
۱۳٫۲. بررسی DNS از طریقSOCKS5 ..... ۱۰۴
۱۳٫۳. بررسیTLS ۱۰۴
۱۴. عیب‌یابی مشکلاتProxy ۱۰۵
بخش دهم ۱۰۶
آموزش تنظیم NTPو سرورهای زمان در ایران در Linuxو Windows ۱۰۷
۱. مقدمه ۱۰۷
۲. NTP چگونه کار می‌کند؟ ۱۰۷
۳. سرورهای زمانی داخل ایران ۱۰۸
فهرست سرورهای پیشنهادی برای آزمایش ۱۰۸
۴. آزمایش دسترسی به سرور NTP ۱۰۹
۵. تنظیم NTPدر Debianو Ubuntuبا Chrony ۱۰۹
۶. تنظیم NTPدر Debianو Ubuntuبا systemd-timesyncd ۱۱۱
۷. تنظیم NTPدر Fedora، Rocky Linuxو AlmaLinux ۱۱۲
۸. تنظیم NTPدر Arch Linuxو Manjaro ۱۱۲
۹. تنظیم ساعت و NTPدر Windows 10و Windows 11 ..... ۱۱۳
۱۰. تنظیم NTPدر Windows Server ۱۱۴
۱۱. تنظیم منطقه زمانی ایران ((Asia/Tehran ۱۱۵
۱۲. تنظیم NTPدر MikroTik ۱۱۶
۱۳. ساخت سرور NTPداخلی برای شبکه سازمانی ۱۱۶

۱۴. عیب‌یابی مشکلات NTP ۱۱۷
۱۵. نکات امنیتی و عملیاتی ۱۱۸
۱۶. جمع‌بندی دستورات مهم ۱۱۸
بخش یازدهم ۱۱۹
آسان‌دسک؛ نرم‌افزار دسترسی و پشتیبانی از راه دور ۱۲۰
آموزش تنظیم آسان‌دسک برای اتصال به سرور ایران ۱۲۰
نرم‌افزار RustDesk ؛ کنترل از راه دور با امکان استفاده از سرور اختصاصی ۱۲۲
معرفی RustDesk ۱۲۲
کاربردهای RustDesk ۱۲۲
قابلیت‌های کلیدی RustDesk ۱۲۳
۱. اتصال از راه دور ۱۲۳
۲. پشتیبانی از سیستم‌عامل‌های مختلف ۱۲۳
۳. سرور اختصاصی ۱۲۳
۴. انتقال فایل و کلیپ‌بورد ۱۲۳
۵. تنظیم کیفیت تصویر ۱۲۳
۶. پشتیبانی از چند نمایشگر ۱۲۳
دانلود RustDesk برای ویندوز ۱۲۳
تنظیم RustDesk برای استفاده از سرور اختصاصی ۱۲۴
نکات امنیتی ۱۲۴

بخش اول

لیست فایل‌هایISO لینوکسی قابل دسترس در سرورهای داخلی ایران

فایل‌های ISOنسخه‌های آماده و قابل بوت سیستم‌عامل‌ها و توزیع‌های لینوکس هستند که برای نصب یا اجرای آزمایشی سیستم‌عامل استفاده می‌شوند. این فایل‌ها معمولاً شامل تصویر کامل دیسک نصب، بسته‌های موردنیاز نصب سیستم‌عامل و ابزارهای راه‌اندازی هستند.

برای دریافت فایل‌های ISO، کاربران معمولاً از وب‌سایت رسمی توزیع لینوکس یا سرورهای آینه (Mirror Server) استفاده می‌کنند. سرورهای Mirrorنسخه‌ای از فایل‌ها و مخازن اصلی را در زیرساخت دیگری نگهداری می‌کنند تا کاربران بتوانند آن‌ها را از مسیر شبکه‌ای نزدیک‌تر دریافت کنند.

فهرست سرورهای داخلی برای دریافت فایل‌های ISO

در فهرست زیر تعدادی از سرورهای ایرانی معرفی شده‌اند که می‌توان برای بررسی فایل‌های نصب توزیع‌های لینوکس و سایر منابع نرم‌افزاری از آن‌ها استفاده کرد.

سرور	نشانی
Mobinhost	https://mirror.mobinhost.com/
IranServer	https://mirror.iranserver.com
دانشگاه صنعتی اصفهان (IUT)	https://repo.iut.ac.ir/repo/
Sindad Cloud – Ubuntu	https://ir.ubuntu.sindad.cloud/releases/
FamaServer	https://mirror.famaserver.com/

جدول زیر مجموعه‌ای از مخازن نرم‌افزاری و سرورهای آینه xMirrorمستقر در ایران را معرفی می‌کند که توسط دانشگاه‌ها، شرکت‌های ارائه‌دهنده خدمات اینترنتی، شرکت‌های خصوصی و سازمان‌های دولتی میزبانی می‌شوند. این مخازن نسخه‌هایی از بسته‌های نرم‌افزاری و آرشیو توزیع‌های مختلف لینوکس را نگهداری می‌کنند و امکان دریافت بسته‌ها و به‌روزرسانی سیستم‌عامل را از طریق زیرساخت داخلی فراهم می‌سازند.

هدف از تهیه این جدول، ارائه یک مرجع کاربردی برای مدیران سیستم، توسعه‌دهندگان و کاربران لینوکس است تا بتوانند متناسب با توزیع مورد استفاده خود، مخزن مناسب را شناسایی و از آن برای دریافت فایل‌های ISO، بسته‌های نرم‌افزاری و به‌روزرسانی‌های سیستم استفاده کنند.

سایت مخزن (دامنه .ir)	مخزن/ISO لینک‌های دانلود	تماس/پشتیبانی
repo.iut.ac.ir **(دانشگاه صنعتی اصفهان)**	https://repo.iut.ac.ir/repo/	ITCenter (iut) ؛ ایمیل: itcenter@of.iut.ac.ir
mirror.arvancloud.ir	http://mirror.arvancloud.ir	ArvanCloud
mirror.shatel.ir	http://mirror.shatel.ir/debian/	**shatel.ir** شرکت شاتل؛ وب‌سایت (پشتیبانی)
archive.ubuntu.petiak.ir	https://archive.ubuntu.petiak.ir/ubuntu/	
archive.ito.gov.ir	https://archive.ito.gov.ir/ubuntu/	سازمان فناوری اطلاعات ایران (وب‌سایت ITO)

بخش دوم

لیست dns های جایگزین

DNSمثل «دفترچه تلفن اینترنت» است که نام سایت‌ها مثل example.irرا به آدرس IP تبدیل می‌کند .DNS داخلی یعنی از سرورهای DNS مستقر در داخل ایران استفاده کنید، نهDNS های خارجی مثل ۸،۸،۸،۸ (گوگل).

۱ .**سرعت بیشتر** :چون درخواست‌های DNS مسیر کوتاه‌تری داخل شبکه ملی طی می‌کنند، پاسخ سریع‌تر دریافت می‌شود و زمان بارگذاری صفحات کم‌تر می‌شود.

۲ .**پایداری دسترسی به سرویس‌های ایرانی** :بعضی سایت‌ها و سرویس‌های ایرانی ممکن است رویDNS های خارجی درست resolve نشوند یا کند باشند .DNS داخلی این مشکل را حل می‌کند و دسترسی پایدارتر می‌شود.

DNS های پیشنهادی اپراتورهای اینترنت

۱.شرکت مخابرات ایران برای مشترکان خود DNS زیر را پیشنهاد کرده است ، 5.200.200.200آدرس DNS اولیه (Primary) پیشنهادی مخابرات ایران برای کاربران اینترنت پرسرعت این شرکت است.

بر اساس گزارش‌ها مخابرات ایران این DNS را عمدتاً در شرایط **محدودیت شبکه بین‌المللی یا اختلال در دسترسی** توصیه کرده است. هدف رسمی اعلام شده این است:

- **بهینه‌سازی مسیریابی داخلی** :هدایت درخواست‌های کاربران در شبکه داخلی ایران، بهبود سرعت و پایداری دسترسی به وب‌سایت‌ها و اپلیکیشن‌های داخلی (مانند خدمات دولتی، آموزشی و غیره).
- **کاهش وابستگی به مسیریابی بین‌المللی** :در شرایط اختلال در شبکه بین‌المللی، جلوگیری از تأثیرگذاری ترجمه DNS بر دسترسی به خدمات داخلی

مخابرات 5.200.200.200

۲.همچنین سازمان نظام صنفی رایانه‌ای کشور در راهنمای خود، DNSهای زیر را برای اپراتورهای مختلف اعلام کرده است :

DNS های داخلی ایران را بر اساس **نوع کاربرد** به دو دسته اصلی تقسیم کرده‌ام: یکی **DNSرسمی اپراتورها** که برای دسترسی به وب‌سایت‌های داخلی طراحی شده و دیگری **DNSهای شخص ثالث** که برای بهینه‌سازی استفاده می‌شوند. در زیر جدول تفکیک شده آمده است:

جدول تفکیکDNS های داخلی ایران

نام سرویس/اپراتور	DNSاول	DNSدوم	کاربرد اصلی/توضیحات
دسته اول DNS : رسمی اپراتورها (دسترسی پایدار به داخل)			
مخابرات(TCI)	217.218.155.155	217.218.127.127	DNSرسمی پیشنهادی سازمان نظام صنفی رایانه‌ای برای دسترسی به وب‌سایت‌های داخلی
مخابرات (گزینه جایگزین)	5.200.200.200	—	DNSجایگزین اعلام شده توسط مخابرات که در فهرست رسمی نیز ذکر شده است
شاتل	85.15.1.14	85.15.1.15	DNSرسمی پیشنهادی شاتل
پیشگامان	5.202.100.101	—	DNSرسمی پیشنهادی پیشگامان
وندا	185.164.72.97	—	DNSرسمی پیشنهادی وندا
دی ان اس پرو	87.107.110.109	—	DNSرسمی پیشنهادی سرویس DNS Pro
ایرانسل	5.200.200.200	—	DNSرسمی پیشنهادی ایرانسل
همراه اول	194.225.152.14	—	DNSرسمی پیشنهادی همراه اول
مرکز اطلاعات (NIC (شبکه ایران	194.225.152.10	—	DNSمرتبط با مرکز مدیریت اینترنت ایران
دسته دوم DNS :شخص ثالث (دور زدن تحریم/بهینه‌سازی بازی)			
الکترو(Electro)	78.157.42.100	78.157.42.101	یکی ازDNS های شناخته‌شده برای دور زدن تحریم‌ها و بهبود اتصال بازی‌ها(PS) ، Xbox، (PCکه در فهرست رسمی نیز ذکر شده است
شکن(Shecan)	178.22.122.100	185.51.200.2	قدیمی‌ترین و رایج‌ترین DNS داخلی برای دور زدن تحریم‌ها و دسترسی به وب‌سایت‌های خارجی
۴۰۳ آنلاین(403)	10.202.10.202	10.202.10.102	ارائه‌دهنده DNS برای دور زدن تحریم‌ها، پشتیبانی از DoH/DoT آدرس خصوصی، فقط قابل استفاده در داخل ایران
رادار گیم(Radar)	10.202.10.10	10.202.10.11	مخصوص بازی‌ها (با پشتیبانی بنیاد ملی بازی‌های رایانه‌ای)، برای کاهش پینگ در بازی‌های پشتیبانی شده (آدرس خصوصی، فقط قابل استفاده در داخل ایران)

بگذر(Begzar)	185.55.226.26	185.55.225.25	گزینه‌ای برای دسترسی به وب‌سایت‌ها یا سرویس‌هایی که کاربران ایرانی را محدود کرده‌اند

۳.دسته سوم DNS :شخص ثالث

نام سرویس / اپراتور	DNSاول	DNSدوم	کاربرد اصلی / توضیحات
مخابرات	217.218.155.155	217.218.127.127	پیشنهادی اپراتور مخابرات برای دسترسی پایدار به سایت‌های داخلی DNS
مخابرات (پیشنهاد دیگر)	5.200.200.200	—	جایگزین معرفی‌شده توسط مخابرات DNS
شاتل	85.15.1.14	85.15.1.15	پیشنهادی اپراتور شاتل DNS
پیشگامان	5.202.100.101	—	پیشنهادی اپراتور پیشگامان DNS
وناد	185.164.72.97	—	پیشنهادی اپراتور وناد DNS
دی ان اس پرو	87.107.110.109	—	پیشنهادی سرویس دی ان اس پرو DNS
ایرانسل	5.200.200.200	—	پیشنهادی اپراتور ایرانسل DNS
همراه اول	194.225.152.14	—	پیشنهادی اپراتور همراه اول DNS
NIC	194.225.152.10	—	(NIC) مرتبط با مرکز مدیریت اینترنت ایران DNS
الکترو Electro	78.157.42.100	78.157.42.101	یکی از شناخته‌شده‌ترین DNS ها برای دور زدن تحریم‌ها و بهبود اتصال در بازی‌هاPS ، Xbox، PC
شکن Shecan	178.22.122.100	185.51.200.2	قدیمی‌ترین و رایج‌ترین سرویس داخلی برای عبور از تحریم وب‌سایت‌ها و سرویس‌های خارجی
۴۰۳ آنلاین 403	10.202.10.202	10.202.10.102	ارائه‌دهنده DNS برای دور زدن تحریم‌ها؛ از پروتکل‌های DoH و DoT نیز پشتیبانی می‌کند (آدرس خصوصی، فقط داخل ایران)
رادار گیم Radar	10.202.10.10	10.202.10.11	مخصوص گیمرها (با پشتیبانی بنیاد ملی بازی‌های رایانه‌ای)؛ برای کاهش (پینگ در بازی‌های پشتیبانی‌شده (آدرس خصوصی، فقط داخل ایران
بگذر Begzar	185.55.226.26	185.55.225.25	گزینه‌ای مکمل برای باز کردن سایت‌ها و سرویس‌هایی که کاربران ایرانی را محدود کرده‌اند

۴.DNSهای مرتبط با مؤسسات دانشگاهی و پژوهشی ایران

نام مؤسسه	DNSاول	DNSدوم	توضیحات
دانشگاه تهران TehranUniv	194.225.62.80	194.225.62.81	آدرس DNS مرتبط با دانشگاه علوم پزشکی تهران، در چندین فهرست ذکر شده است
دانشگاه صنعتی شریف SUT	194.225.0.10	194.225.0.11	آدرس DNS دانشگاه صنعتی شریف SUT)، ذکر شده در فهرست DNS-MNS
دانشگاه صنعتی امیرکبیر AUT	194.225.64.10	194.225.64.11	آدرس DNS دانشگاه صنعتی امیرکبیر AUT)
دانشگاه صنعتی اصفهان IUT	176.101.48.10	176.101.48.11	آدرس DNS دانشگاه صنعتی اصفهان IUT)

دانشگاه فردوسی مشهد FerdowsiUniv	46.143.212.42	—	آدرس DNS دانشگاه فردوسی مشهد
پژوهشگاه دانش‌های بنیادی IPM	194.225.152.10	194.225.152.12	آدرس DNS مرتبط با پژوهشگاه دانش‌های بنیادیIPM)، که در تست‌ها بالایی داشته است

۵.جدولDNS های پیشنهادی (از داخل ایران)

نام سرویس	DNSاول	DNSدوم	نوع / توضیحات
DynX AntiBan	10.139.177.18	10.139.177.16	**داخلی** (**تونل ایران**)؛ به‌طور خاص برای تغییر موقعیت جغرافیایی (GeoLocation) و دور زدن تحریم‌ها طراحی شده است .
DynX(نسخه قدیمی‌تر)	10.70.95.150	10.70.95.162	**داخلی**؛ آدرس‌های جایگزین که در برخی اسکریپت‌ها و لیست‌ها ذکر شده‌اند .
Vanilla DNS	10.139.177.21	10.139.177.22	**داخلی**؛ یک سرویس «هوشمند و سریع برای دور زدن تحریم» از ایران که در اپلیکیشن رسمی آن از پیش تنظیم شده است .
Cloudflare	1.1.1.1	1.0.0.1	**خارجی** (**عمومی**)؛ به عنوان یک DNS سریع و پایدار جهانی شناخته می‌شود، اما در شرایط خاص ایران ممکن است به تنهایی برای دور زدن تحریم‌ها کافی نباشد و نیاز به ترکیب با روش‌های دیگر داشته باشد .
Google Public DNS	8.8.8.8	8.8.4.4	**خارجی** (**عمومی**)؛ یکی از شناخته‌شده‌ترینDNS های جهانی. برخی کاربران گزارش داده‌اند که در برخی سناریوها (مانند دسترسی به سرویس‌های خاص) مؤثر بوده است، اما در ایران ممکن است با محدودیت مواجه شود .

بخش سوم

روش تظیم dns در ویندوز

پس از انتخاب DNS مناسب از جدول‌های بالا، مراحل زیر را در ویندوز انجام دهید :

۱. **باز کردن تنظیمات شبکه :**کلیدهای Win + Rرا همزمان فشار دهید، عبارت ncpa.cplرا تایپ کرده و Enter بزنید.
۲. **انتخاب آداپتور :**روی اتصال اینترنت خود Wi-Fi یا Ethernet راست‌کلیک کرده و **Properties**را انتخاب کنید.
۳. **انتخاب پروتکل :**در لیست، گزینه **Internet Protocol Version 4 (TCP/IPv4)**را پیدا کرده و روی آن دابل‌کلیک کنید (یا انتخاب کرده و Properties را بزنید.
۴. **وارد کردن DNS:** گزینه **Use the following DNS Server addresses**را فعال کنید. سپس آدرس **Preferred DNS**و **Alternate DNS**را از جدول‌های بالا وارد کنید.
۵. **ذخیره :**روی OK کلیک کنید تا تنظیمات اعمال شود.

پاک کردن کش DNS مهم :پس از تغییرDNS ، بهتر است کش سیستم را پاک کنید تا آدرس‌های قدیمی حذف شوند.
در منوی Start عبارت cmdرا جستجو کرده، Command Promptرا باز کنید و دستور زیر را اجرا کنید :

```
ipconfig /flushdns
```

روش تنظیم DNS در لینوکس

در ابتدا روش های خالی کردن کش dns

برای پاک کردن کش DNS در لینوکس، ابتدا باید بررسی کنید که سیستم شما از کدام سرویس کش استفاده می‌کند، زیرا دستورات بر اساس سرویس متفاوت است. اکثر توزیع‌های مدرن مانندUbuntu ، Fedora، Debianبه طور پیش‌فرض از systemd-resolvedاستفاده می‌کنند.

تشخیص سرویس کشDNS

ابتدا سرویس فعال را بررسی کنید:

 # بررسی وضعیت سرویس‌ها

```
systemctl status systemd-resolved
systemctl status nscd
systemctl status dnsmasq
```

بررسی فایلresolv.conf

```
cat /etc/resolv.conf
```

اگر nameServer 127.0.0.53را مشاهده کردید، یعنی از systemd-resolvedاستفاده می‌کنید .

دستورات پاک کردن کش بر اساس سرویس

سرویس	دستور پاک کردن کش
systemd-resolved (رایج‌ترین)	sudo resolvectl flush-caches
nscd (سیستم‌های قدیمی‌تر)	sudo systemctl restart nscd
dnsmasq	sudo systemctl restart dnsmasq
NetworkManager (همراه با dnsmasq)	sudo systemctl restart NetworkManager

نکته :در نسخه‌های قدیمی‌ترsystemd ، دستور sudo systemd-resolve --flush-cachesاست، اما در نسخه‌های جدید توصیه می‌شود از resolvectlاستفاده کنید .

بررسی نتیجه پاک کردن

پس از اجرا، می‌توانید با دستور زیر تأیید کنید که کش خالی شده است:

```
resolvectl statistics | grep "Current Cache Size"
```

باید Current Cache Size: 0را نشان دهد .

خالی کردن کش dns مرورگرها

مرورگرها مانند Chrome و Firefox کش DNS داخلی خود را دارند که مستقل از سیستم است و پاک کردن کش سیستم لزوماً آن را پاک نمی‌کند. در زیر روش‌های پاک کردن کش DNS مرورگرهای اصلی آمده است:

Google Chrome / Chromium / Edge هسته Chromium)

مرورگرهای مبتنی بر Chromium از یک صفحه عیب‌یابی داخلی برای پاک کردن کش DNS استفاده می‌کنند:

۱. یک صفحه جدید باز کنید و در نوار آدرس وارد کنید

```
chrome://net-internals/#dns
```

۲. روی دکمه **"Clear host cache"**(پاک کردن کش میزبان) کلیک کنید .

توجه :این عملیات فقط کش DNS مرورگر را پاک می‌کند. اگر مشکل همچنان ادامه داشت، می‌توانید به تب Socketsبروید یا مستقیماً chrome://net-internals/#socketsرا باز کنید) و روی **"Flush socket pools"**کلیک کنید تا اتصالات موجود بسته شوند .

Mozilla Firefox

روش پاک کردن کش DNS در فایرفاکس از طریق صفحه تنظیمات شبکه داخلی انجام می‌شود:

۱. یک صفحه جدید باز کنید و در نوار آدرس وارد کنید

```
about:networking#dns
```

۲. روی دکمه **"Clear DNS Cache"** پاک کردن کشDNS کلیک کنید .

مرورگرهای دیگر

- **Safari** مک‌اواس معمولاً از طریق ترمینال sudo killall -HUP mDNSResponderکش سیستم را پاک می‌کند؛ Safariمعمولاً کش DNS مستقلی ندارد که نیاز به پاک کردن جداگانه داشته باشد.
- **مرورگرهای قدیمی‌تر یا روش‌های جایگزین** :برخی منابع اشاره می‌کنند که بستن کامل مرورگر و باز کردن مجدد آن نیز می‌تواند کش DNS در حافظه را آزاد کند به‌ویژه Chrome

بخش چهارم

تنظیمات dns جدید در لینوکس

روش‌های موقت و دائمی به تفکیک فایل و دستورات عملیاتی ارائه شده است. Dns مورد نظر خود را در مسیر های داده شده زیر جایگزین کنید.

اوبونتو Ubuntu 26.04/22.04/24.04)

ابونتو مدرن به طور پیش‌فرض از **Netplan** به عنوان رابط پیکربندی شبکه و در پشت صحنه از systemd-resolved برای مدیریت DNS استفاده می‌کند.

۱.تنظیم موقت (با ری‌استارت از بین می‌رود)

ویرایش مستقیم فایل :resolv.conf

```
sudo vim /etc/resolv.conf
```

تغییرات به شرح زیر است (اگر خط nameServer اصلی وجود دارد، آن را حذف یا کامنت کنید):

```
nameServer 8.8.8.8
nameServer 1.1.1.1
```

این کار بلافاصله اعمال می‌شود، اما پس از ری‌استارت یا راه‌اندازی مجدد سرویس شبکه توسط سیستم بازنویسی خواهد شد .

۲.تنظیم دائمی روش توصیه شده Netplan :

ابتدا فایل پیکربندی Netplan را پیدا کنید (معمولاً در / /etc/netplan/با نامی شبیه:

```
50-cloud-init.yaml  یا01-netcfg.yaml
```

```
sudo vim /etc/netplan/50-cloud-init.yaml
```

در بخش کارت شبکه مربوطه مانند ethernets ، فیلد nameserversرا اضافه یا تغییر دهید:

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: true
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

ذخیره کنید و سپس اعمال کنید:

```
sudo netplan apply
```

این روش پس از ری‌استارت نیز دائمی خواهد بود .

دبیان بسته به اینکه محیط دسکتاپ باشد یا سرور، روش‌های متفاوتی دارد. اگر systemd-resolvedنصب شده باشد، ویرایش مستقیم resolv.confنیز ممکن است بازنویسی شود.

۱.تنظیم موقت

ابتدا ویرایش فایل تنظیمات زیر:

```
/etc/resolv.conf
```

ویرایش:

```
sudo vim /etc/resolv.conf
```

خط nameServer را اضافه کنید. توجه: اگر سیستم شما از systemd-resolvedیا NetworkManagerاستفاده می‌کند،

```
nameServer 8.8.8.8
nameServer 1.1.1.1
```

این فایل ممکن است یک لینک نمادین باشد که به طور خودکار مدیریت می‌شود و تغییرات ممکن است پس از ری‌استارت از بین برود .

۲.تنظیم دائمی سرور: پیکربندی سنتیinterfaces

اگر از NetworkManager استفاده نمی‌کنید، فایل / etc/network/interfacesرا ویرایش کنید:

```
sudo vim /etc/network/interfaces
```

در بخش پیکربندی کارت شبکه، خط dns-nameserversرا اضافه کنید:

```
auto eth0
iface eth0 inet static
   address 192.168.1.100
   netmask 255.255.255.0
   gateway 192.168.1.1
   dns-nameservers 8.8.8.8 1.1.1.1
```

پس از ذخیره، سرویس شبکه را ری‌استارت کنید:

```
sudo systemctl restart networking
```

تنظیم دائمی DNSدر دبیان با استفاده از NetworkManager

در سیستم‌عامل Debian، اگر مدیریت اتصال شبکه بر عهده سرویس NetworkManagerباشد، می‌توانید با استفاده از ابزار خط فرمان nmcli، سرورهای DNSرا به‌صورت دائمی تنظیم کنید. این روش برای اتصالات کابلی Ethernetو شبکه‌های بی‌سیم Wi-Fiکاربرد دارد و تنظیمات ذخیره‌شده پس از راه‌اندازی مجدد سیستم نیز باقی می‌مانند.

۱. بررسی وضعیت NetworkManager

ابتدا وضعیت سرویس را بررسی کنید:

```
systemctl status NetworkManager
```

اگر سرویس نصب و فعال باشد، عبارت active (running)نمایش داده می‌شود. در صورت نصب نبودن NetworkManager، می‌توانید آن را نصب و فعال کنید:

```
sudo apt update
sudo apt install network-manager
sudo systemctl enable --now NetworkManager
```

توجه داشته باشید که اگر سیستم از مدیر شبکه دیگری استفاده می‌کند، پیش از تغییر سرویس مدیریت شبکه، تنظیمات فعلی را بررسی کنید.

۲. شناسایی نام اتصال شبکه

برای مشاهده اتصالات ذخیره‌شده اجرا کنید:

```
nmcli connection show
```

نمونه خروجی:

```
NAME                UUID                                  TYPE      DEVICE
Wired connection 1  xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx  ethernet  enp3s0
Home-WiFi           yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy  wifi      --
```

در این خروجی، ستون NAMEنام پروفایل اتصال را مشخص می‌کند. نام اتصال ممکن است با نام رابط شبکه، مانند enp3s0، متفاوت باشد. در دستورات زیر باید نام واقعی اتصال خود را جایگزین Wired connection 1کنید.

۳. تنظیم DNSبه‌صورت دائمی

در این مثال، دو سرور DNSعمومی Googleو Cloudflareتنظیم می‌شوند:

```
Google Public DNS: 8.8.8.8
Cloudflare DNS: 1.1.1.1
```

ابتدا آدرس‌های DNSرا برای اتصال ذخیره کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.dns "8.8.8.8 1.1.1.1"
```

سپس از دریافت و استفاده از DNSهای خودکار ارائه‌شده توسط DHCP جلوگیری کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.ignore-auto-dns yes
```

در پایان، اتصال را مجدداً فعال کنید تا تنظیمات جدید اعمال شوند:

```
sudo nmcli connection up "Wired connection 1"
```

با اجرای این دستورات، DNSهای مشخص‌شده در پروفایل اتصال ذخیره می‌شوند و معمولاً پس از راه‌اندازی مجدد سیستم نیز باقی می‌مانند.

نکته :اجرای دستور آخر ممکن است ارتباط شبکه را برای مدت کوتاهی قطع کند. اگر از طریق SSHبه سیستم متصل هستید، این احتمال را در نظر بگیرید.

۴. تنظیم DNSبرای Wi-Fi

اگر از شبکه بی‌سیم استفاده می‌کنید، ابتدا نام اتصال را با دستور nmcli connection showپیدا کنید. سپس، برای نمونه با فرض نام Home-WiFi، دستورات زیر را اجرا کنید:

```
sudo nmcli connection modify "Home-WiFi" ipv4.dns "8.8.8.8 1.1.1.1"
sudo nmcli connection modify "Home-WiFi" ipv4.ignore-auto-dns yes
sudo nmcli connection up "Home-WiFi"
```

۵. بررسی تنظیمات و آزمایش DNS

برای اطمینان از ذخیره تنظیمات، اجرا کنید:

```
nmcli connection show "Wired connection 1" | grep -E 'ipv4.dns:|ipv4.ignore-auto-dns:'
```

مقادیر مورد انتظار عبارت‌اند از:

```
ipv4.dns:                       8.8.8.8,1.1.1.1
ipv4.ignore-auto-dns:           yes
```

برای آزمایش حل نام دامنه نیز از دستور زیر استفاده کنید:

```
getent ahosts debian.org
```

اگر آدرس‌های IPنمایش داده شوند، سیستم توانسته است نام دامنه را به آدرس IPتبدیل کند. در صورت فعال بودن systemd-resolved، می‌توانید جزئیات DNSفعال را نیز با دستور زیر مشاهده کنید:

```
resolvectl status
```

۶. بازگرداندن تنظیمات به حالت خودکار

اگر بخواهید DNSمجدداً از DHCP دریافت شود، دستورات زیر را اجرا کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.dns ""
sudo nmcli connection modify "Wired connection 1" ipv4.ignore-auto-dns no
sudo nmcli connection up "Wired connection 1"
```

در نهایت، برای اطمینان از ماندگاری تنظیمات، سیستم را راه‌اندازی مجدد کنید و پس از اتصال مجدد به شبکه، وضعیت DNSرا دوباره بررسی کنید.

تنظیم دائمی DNSدر دبیان با استفاده از systemd-resolved

در دبیان، سرویس systemd-resolvedوظیفه مدیریت درخواست‌های DNSو تبدیل نام دامنه‌ها به آدرس IPرا بر عهده می‌گیرد. اگر این سرویس در سیستم شما فعال و مسئول مدیریت DNSباشد، می‌توانید سرورهای DNSرا در فایل پیکربندی آن تعریف کنید تا تنظیمات پس از راه‌اندازی مجدد سیستم نیز باقی بمانند.

در این آموزش، DNSهای عمومی Googleو Cloudflareرا تنظیم می‌کنیم.

۱. بررسی وضعیت سرویس

ابتدا بررسی کنید که سرویس نصب و فعال است یا خیر:

```
systemctl status systemd-resolved
```

اگر سرویس فعال باشد، عبارت active (running)نمایش داده می‌شود.

برای بررسی دقیق‌تر وضعیت DNSو رابط‌های شبکه نیز اجرا کنید:

```
resolvectl status
```

اگر سرویس وجود ندارد یا فعال نیست، پیش از ادامه بررسی کنید که سیستم شما از چه روشی برای مدیریت DNSاستفاده می‌کند. در دبیان، استفاده از systemd-resolvedبه‌صورت پیش‌فرض در همه نصب‌ها تضمین‌شده نیست.

۲. ویرایش فایل پیکربندی

فایل اصلی تنظیمات سرویس در مسیر زیر قرار دارد:

```
/etc/systemd/resolved.conf
```

برای ویرایش آن با ویرایشگر Vimاجرا کنید:

```
sudo vim /etc/systemd/resolved.conf
```

اگر Vimنصب نیست، می‌توانید از ویرایشگر Nano استفاده کنید:

```
sudo nano /etc/systemd/resolved.conf
```

در فایل، بخش [Resolve]را پیدا کنید. اگر وجود ندارد، آن را اضافه کنید. سپس خط DNS=را به شکل زیر تنظیم کنید:

[Resolve]

```
DNS=8.8.8.8 1.1.1.1
```

در این تنظیم:

- DNS=8.8.8.8سرور Google Public DNSرا مشخص می‌کند.

- DNS=1.1.1.1سرور Cloudflareرا مشخص می‌کند.

اگر خط DNS=از قبل وجود دارد، مقدار آن را اصلاح کنید و از تعریف چندباره و متناقض این گزینه در بخش‌های مختلف فایل خودداری کنید.

در Vim، برای ذخیره و خروج، کلید Escرا فشار دهید و دستور زیر را وارد کنید:

```
:wq
```

در Nano، کلیدهای Ctrl + Oو سپس Enterرا برای ذخیره و Ctrl + Xرا برای خروج فشار دهید.

۳. راه‌اندازی مجدد سرویس

پس از ذخیره فایل، سرویس را مجدداً راه‌اندازی کنید:

```
sudo systemctl restart systemd-resolved
```

سپس وضعیت آن را بررسی کنید:

```
systemctl status systemd-resolved
```

برای مشاهده DNSهای فعال:

```
resolvectl status
```

در خروجی، بخش Globalیا اطلاعات مربوط به رابط شبکه را بررسی کنید تا مشخص شود DNSهای موردنظر اعمال شده‌اند.

تنظیم DNS در فدورا Fedora 40 ، ۴۱ و ۴۲

۱. تنظیم موقت DNS

فایل / etc/resolv.confرا می‌توان برای بررسی یا تغییر DNS ویرایش کرد:

```
sudo vim /etc/resolv.conf
```

برای نمونه، خطوط زیر را اضافه کنید یا مقادیر موجود را تغییر دهید:

```
nameServer 8.8.8.8
nameServer 1.1.1.1
```

توجه :در پیکربندی معمول فدورا، این فایل توسط NetworkManager یا سازوکار مدیریت DNS سیستم تولید یا به‌روزرسانی می‌شود. بنابراین، تغییر مستقیم آن ممکن است با فعال‌سازی مجدد اتصال یا راه‌اندازی مجدد سیستم از بین برود. برای تنظیم دائمی، روش زیر را به کار ببرید.

۲. تنظیم دائمی DNS باNetworkManager

ابتدا نام اتصال شبکه را پیدا کنید:

```
nmcli connection show
```

فرض می‌کنیم نام اتصال Wired connection 1است. اگر نام دیگری مشاهده کردید، آن را جایگزین کنید.

سرورهای DNS را تنظیم کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.dns "8.8.8.8 1.1.1.1"
```

برای جلوگیری از استفاده ازDNS های دریافتی ازDHCP ، دستور زیر را اجرا کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.ignore-auto-dns yes
```

سپس اتصال را دوباره فعال کنید:

```
sudo nmcli connection up "Wired connection 1"
```

این روش برای اتصالات Ethernet و Wi-Fi تحت مدیریت NetworkManager کاربرد دارد. توجه کنید که فعال‌سازی مجدد اتصال ممکن است ارتباط شبکه را برای چند لحظه قطع کند.

۳. بررسی تنظیمات

برای مشاهدهDNS های ذخیره‌شده اجرا کنید:

```
nmcli connection show "Wired connection 1" | grep -E 'ipv4.dns:|ipv4.ignore-auto-dns:'
```

برای آزمایش حل نام دامنه نیز دستور زیر را اجرا کنید:

```
getent ahosts fedoraproject.org
```

پس از راه‌اندازی مجدد سیستم، می‌توانید این آزمایش‌ها را تکرار کنید تا از ماندگاری تنظیمات مطمئن شوید.

نکته :این تنظیمات مربوط به IPv4 هستند. اگر سیستم از IPv6 نیز استفاده می‌کند، تنظیمات DNS مربوط به IPv6 را جداگانه بررسی کنید. همچنین، برای بازگشت به DNS خودکار، مقدار ipv4.dnsرا پاک و ipv4.ignore-auto-dnsرا روی noتنظیم کنید.

تنظیم DNS درopenSUSE (Leap / Tumbleweed)

درopenSUSE ، مدیریت شبکه معمولاً با یکی از دو روش NetworkManagerیا Wickedانجام می‌شود. NetworkManagerبیشتر در سیستم‌های دسکتاپ و Wicked در برخی نصب‌های سروری استفاده می‌شود. روش صحیح تنظیم DNS به مدیر شبکه فعال بستگی دارد.

۱. شناسایی روش مدیریت شبکه

برای بررسی وضعیت سرویس‌ها اجرا کنید:

```
systemctl is-active NetworkManager
systemctl is-active wicked
```

همچنین می‌توانید اتصال‌های تحت مدیریت NetworkManager را مشاهده کنید:

```
nmcli connection show
```

اگر NetworkManager فعال است، روش اول را دنبال کنید. اگر سیستم با Wicked مدیریت می‌شود، از روش YaST استفاده کنید.

۲. تنظیم موقت DNS

برای مشاهده فایل DNS سیستم اجرا کنید:

```
cat /etc/resolv.conf
```

در صورت نیاز می‌توانید فایل را با ویرایشگر Vim باز کنید:

```
sudo vim /etc/resolv.conf
```

برای نمونه، خطوط زیر را وارد کنید:

```
nameServer 8.8.8.8
nameServer 1.1.1.1
```

این روش فقط زمانی مناسب است که فایل به‌صورت مستقل مدیریت شود. اگر NetworkManager یا سازوکار دیگری این فایل را مدیریت کند، تغییرات ممکن است بازنویسی شوند؛ بنابراین برای تنظیم دائمی، روش متناسب با مدیر شبکه را انتخاب کنید.

۳. تنظیم دائمی DNS با NetworkManager

ابتدا نام اتصال را پیدا کنید:

```
nmcli connection show
```

فرض می‌کنیم نام اتصال Wired connection 1 است. اگر نام دیگری دارید، آن را جایگزین کنید.

سرورهای DNS را برای IPv4 تنظیم کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.dns "8.8.8.8 1.1.1.1"
```

برای نادیده گرفتن DNS های خودکار دریافتی از DHCP:

```
sudo nmcli connection modify "Wired connection 1" ipv4.ignore-auto-dns yes
```

سپس اتصال را دوباره فعال کنید:

```
sudo nmcli connection up "Wired connection 1"
```

این تنظیمات در پروفایل اتصال ذخیره می‌شوند و پس از راه‌اندازی مجدد سیستم نیز باقی می‌مانند.

آیا لازم است rc-manager را روی file قرار دهیم؟

معمولاً خیر. گزینه rc-manager نحوه مدیریت فایل /etc/resolv.conf را مشخص می‌کند و به‌تنهایی روش تنظیم DNS پروفایل اتصال نیست.

اگر دلیل مشخصی برای تغییر این گزینه دارید، ابتدا فایل پیکربندی را بررسی کنید:

```
sudo vim /etc/NetworkManager/NetworkManager.conf
```

در صورت نیاز، زیر بخش [main]مقدار زیر را تنظیم کنید:

```
[main]
rc-manager=file
```

اگر بخش [main]از قبل وجود دارد، بخش تکراری ایجاد نکنید.

پس از تغییر پیکربندی، NetworkManagerرا بارگذاری مجدد کنید:

```
sudo systemctl reload NetworkManager
```

هشدار :تغییر rc-managerبرای همه سیستم‌ها ضروری نیست و ممکن است با شیوه مدیریت فعلی / etc/resolv.conf سازگاری نداشته باشد. پیش از اعمال آن، پیوند نمادین فایل و وضعیت سرویس‌های مدیریت DNS را بررسی کنید.

۴. تنظیم دائمی DNS با YaST وWicked

اگر سیستم از Wicked استفاده می‌کند، می‌توانید DNS را از طریق ابزار گرافیکی YaST تنظیم کنید.

۱. ابزار YaST را از منوی سیستم باز کنید یا در صورت نصب بودن، اجرا کنید:

```
sudo yast2
```

۲. وارد بخش System → Network Settingsیا بخش تنظیمات شبکه متناظر با نسخه نصب‌شده شوید.

۳. زبانه Hostname/DNSرا باز کنید.

۴. آدرس سرورهای DNS را وارد کنید:

```
8.8.8.8
1.1.1.1
```

۵. تغییرات را تأیید و ذخیره کنید.

مسیر دقیق منو ممکن است با نسخه openSUSE و نسخه YaST متفاوت باشد. این روش برای سیستم‌هایی مناسب است که تنظیمات DNS آن‌ها از طریق YaST و سازوکار شبکه سیستم مدیریت می‌شود؛ اگر NetworkManager فعال باشد، ممکن است این بخش برای تنظیم رابط‌های شبکه در دسترس نباشد.

۵. بررسی عملکردDNS

برای بررسی تنظیمات ذخیره‌شده در:NetworkManager

```
nmcli connection show "Wired connection 1" | grep -E 'ipv4.dns:|ipv4.ignore-auto-dns:'
```

برای آزمایش حل نام دامنه:

```
getent ahosts opensuse.org
```

در صورت نمایش آدرس‌هایIP ، حل نام دامنه موفق بوده است. برای بررسی فایل DNS فعال نیز اجرا کنید:

```
cat /etc/resolv.conf
```

محتوای این فایل ممکن است به دلیل استفاده از یک سرویس محلی یا پیوند نمادین، با آدرس سرورهای بالادستی متفاوت باشد.

۶. بازگرداندن DNS به حالت خودکار

اگر از NetworkManager استفاده می‌کنید و می‌خواهید DNS مجدداً از DHCP دریافت شود، اجرا کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.dns ""
sudo nmcli connection modify "Wired connection 1" ipv4.ignore-auto-dns no
sudo nmcli connection up "Wired connection 1"
```

اگر تنظیمات DNS مربوط به IPv6 را نیز تغییر داده‌اید، آن‌ها را جداگانه بررسی و در صورت نیاز به حالت خودکار بازگردانید.

تنظیم DNS در آرچ لینوکس(Arch Linux)

در Arch Linux ، روش تنظیم DNS به سرویس مدیریت شبکه و حل نام دامنه‌ای بستگی دارد که روی سیستم فعال است. برای تنظیم DNS می‌توان از **NetworkManager** یا **systemd-resolved** استفاده کرد. این سرویس‌ها ممکن است به‌صورت مستقل یا در ترکیب با یکدیگر فعالیت کنند؛ بنابراین، پیش از تغییر تنظیمات باید روش مدیریت فعلی شبکه مشخص شود.

۱. بررسی وضعیت شبکه

برای مشاهده رابط‌های شبکه اجرا کنید:

```
ip link
```

برای بررسی اتصالات NetworkManager:

```
nmcli connection show
```

و برای بررسی وضعیت سرویس‌های مرتبط:

```
systemctl is-active NetworkManager
systemctl is-active systemd-resolved
```

اگر از NetworkManager استفاده می‌کنید، روش اول را دنبال کنید. اگر مدیریت DNS بر عهده systemd-resolved است، روش دوم مناسب خواهد بود.

۲. تنظیم موقت DNS

فایل /etc/resolv.conf را باز کنید:

```
sudo vim /etc/resolv.conf
```

برای نمونه، آدرس‌های زیر را وارد کنید:

```
nameServer 8.8.8.8
nameServer 1.1.1.1
```

این مقادیر سرورهای Google Public DNS و Cloudflare را مشخص می‌کنند.

توجه :این روش فقط زمانی مناسب است که فایل به‌صورت مستقل مدیریت شود. اگر فایل توسط NetworkManager ، systemd-resolvedیا سرویس دیگری مدیریت شود، تغییرات ممکن است بازنویسی شوند. برای تنظیم دائمی، بهتر است از روش مدیریت شبکه فعال استفاده کنید.

۳. تنظیم دائمی DNS باNetworkManager

اگر NetworkManager نصب و فعال است، ابتدا نام اتصال را پیدا کنید:

```
nmcli connection show
```

فرض می‌کنیم نام اتصال Wired connection 1است. در صورت متفاوت بودن نام، آن را جایگزین کنید.

سرورهای DNS مربوط به IPv4 را تنظیم کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.dns "8.8.8.8 1.1.1.1"
```

برای نادیده گرفتنDNS های دریافتی از:DHCP

```
sudo nmcli connection modify "Wired connection 1" ipv4.ignore-auto-dns yes
```

سپس اتصال را مجدداً فعال کنید:

```
sudo nmcli connection up "Wired connection 1"
```

این تنظیمات در پروفایل اتصال ذخیره می‌شوند و معمولاً پس از راه‌اندازی مجدد سیستم نیز باقی می‌مانند.

۴. تنظیم دائمی DNS باsystemd-resolved

ابتدا وضعیت سرویس را بررسی کنید:

```
systemctl status systemd-resolved
```

اگر سرویس نصب نشده است، در صورت نیاز آن را نصب کنید:

```
sudo pacman -S systemd-resolved
```

سرویس را فعال کنید:

```
sudo systemctl enable --now systemd-resolved
```

فایل پیکربندی را باز کنید:

```
sudo vim /etc/systemd/resolved.conf
```

در بخش [Resolve]، گزینه DNS=را به شکل زیر تنظیم کنید:

[Resolve]

```
DNS=8.8.8.8 1.1.1.1
```

اگر بخش [Resolve]از قبل وجود دارد، همان بخش را ویرایش کنید و بخش تکراری نسازید.

فایل را ذخیره کنید و سرویس را مجدداً راه‌اندازی کنید:

```
sudo systemctl restart systemd-resolved
```

۵. تنظیم صحیح پیوند نمادین /etc/resolv.conf

برای اینکه برنامه‌های سیستم بتوانند از رابط محلی DNS سرویس systemd-resolved استفاده کنند، معمولاً فایل /etc/resolv.confبه فایل زیر متصل می‌شود:

```
/run/systemd/resolve/stub-resolv.conf
```

ابتدا وضعیت فعلی فایل را بررسی کنید:

```
ls -l /etc/resolv.conf
```

اگر مطمئن هستید که systemd-resolved مسئول مدیریت DNS است و فایل فعلی باید به پیوند استاندارد آن تبدیل شود، اجرا کنید:

```
sudo ln -sf /run/systemd/resolve/stub-resolv.conf /etc/resolv.conf
```

این پیوند باعث می‌شود درخواست‌های DNS سیستم از طریق رابط محلیsystemd-resolved ، معمولاً 127.0.0.53، پردازش شوند.

هشدار :دستور بالا فایل یا پیوند قبلی را جایگزین می‌کند. اگر NetworkManager یا سرویس دیگری مسئول مدیریت DNSباشد، تغییر پیوند بدون بررسی ممکن است باعث اختلال در حل نام دامنه شود.

۶. بررسی عملکردDNS

برای مشاهده وضعیت:DNS

```
resolvectl status
```

برای بررسی تنظیمات فایل:

```
cat /etc/resolv.conf
```

اگر از حالت stub استفاده می‌کنید، معمولاً خط زیر را مشاهده خواهید کرد:

```
nameServer 127.0.0.53
```

این آدرس طبیعی است و به معنای استفاده از رابط محلی systemd-resolved است؛ لزوماً به این معنا نیست که DNSهای عمومی تنظیم نشده‌اند.

برای آزمایش حل نام دامنه:

```
resolvectl query archlinux.org
```

همچنین می‌توانید از دستور زیر استفاده کنید:

```
getent ahosts archlinux.org
```

در صورت نمایش آدرس‌هایIP ، حل نام دامنه موفق بوده است.

۷. بازگرداندن تنظیمات DNS به حالت قبلی

اگر از NetworkManager استفاده می‌کنید و می‌خواهید DNS را مجدداً ازDHCP دریافت کنید:

```
sudo nmcli connection modify "Wired connection 1" ipv4.dns ""
sudo nmcli connection modify "Wired connection 1" ipv4.ignore-auto-dns no
sudo nmcli connection up "Wired connection 1"
```

اگر از systemd-resolved استفاده می‌کنید، می‌توانید گزینه DNS= را در فایل /etc/systemd/resolved.conf حذف یا کامنت کنید و سرویس را دوباره راه‌اندازی کنید:

```
sudo systemctl restart systemd-resolved
```

در این حالت، DNSهای دریافتی از منابع دیگر، مانند تنظیمات رابط شبکه یاDHCP ، می‌توانند مورد استفاده قرار گیرند؛ این رفتار به پیکربندی سیستم بستگی دارد.

نکته پایانی :در Arch Linux یک روش واحد برای تمام نصب‌ها وجود ندارد. برای جلوگیری از تداخل، مشخص کنید کدام سرویس مسئول مدیریت شبکه و DNS است و تنظیمات را مطابق همان روش انجام دهید. همچنین، تنظیم DNS برای IPv4لزوماً DNSهای IPv6 را تغییر نمی‌دهد.

بخش پنجم

جایگزینی مخازن و سرورهای داخلی برای توزیع‌های لینوکس

فهرست جامع مخازن داخلی ایران

مشکلی که کاربران لینوکس اغلب با آن روبه‌رو می‌شوند، از کار افتادن دستور نصب بسته‌ها (مثل apt install یا dnf install)است. دلیل ساده است: به صورت پیش‌فرض، همه توزیع‌های لینوکس بسته‌های خود را از سرورهای خارجی (عمدتاً در اروپا و آمریکا) دانلود می‌کنند. اگر مسیر ارتباط با این سرورها بسته شود، مدیریت بسته‌ها کاملاً متوقف می‌شود.

راه‌حل، استفاده از **مخازن داخلی (Internal Mirrors)** است. مخزن داخلی در واقع یک کپی همگام‌سازی‌شده از مخزن اصلی است که در داخل کشور میزبانی می‌شود. وقتی تنظیمات سیستم را به این مخازن تغییر می‌دهیم، بسته‌ها از داخل ایران دانلود می‌شوند و وابستگی به اینترنت بین‌الملل از بین می‌رود. این کار مزایای دیگری هم دارد: سرعت دانلود بسیار بالاتر، تأخیر کمتر، و پایداری بیشتر در شرایط اختلال شبکه.

فهرست جامع مخازن داخلی ایران

در حال حاضر چندین سرویس‌دهنده ایرانی، مخازن (Mirror) توزیع‌های مختلف لینوکس را به صورت همگام‌سازی‌شده ارائه می‌دهند. هر کدام از این سرویس‌دهنده‌ها ممکن است توزیع‌های خاصی را پوشش دهند و سطح به‌روزرسانی آن‌ها متفاوت باشد.

قسمت اول :ابتدا لیست کلی مخازن

سرویس‌دهنده	آدرس پایه	توزیع‌های پشتیبانی‌شده (نمونه)
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir	Debian, Ubuntu, Mint, Arch, Manjaro, Alpine, Rocky Linux, Fedora, openSUSE
آروان‌کلود ArvanCloud	mirror.arvancloud.ir	Ubuntu, Debian, Arch Linux, CentOS, Alpine
پارس‌پکParsPack	mirror.parspack.com	Ubuntu, Debian (و مخازن زبان‌های برنامه‌نویسی)
ابر فردوسی Ferdowsi Cloud	mirror.ferdowsi.cloud	Ubuntu (و سایر سرویس‌ها)
کَرگَدانKargadan	mirror.kargadan.ir	Fedora, openSUSE
لیاراLiara	linux-mirror.liara.ir	Ubuntu, Debian, Rocky Linux
شاتلShatel	ubuntu.shatel.ir	Ubuntu
میرورهای دانشگاهی	mirrors.sharif.ir شریف mirror.ut.ac.ir (تهران)	Ubuntu

اطلاعات میرور (مخزن) داخلی ایران برای توزیع‌های لینوکس

Debian

ارائه‌دهنده	آدرس میرور
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir
آروان‌کلود	mirror.arvancloud.ir
شاتل	mirror.shatel.ir
ایران‌سرور	mirror.iranserver.com
مبین‌هاست	mirror.mobinhost.com
افرانت	mirror.afranet.com
پردیس	mirrors.pardisco.co
چابکان	mirror2.chabokan.net
آتلانتیس‌کلود	mirror.atlantiscloud.ir
ابر دانا	repo.abrha.net
لیارا	liara.ir/mirrors

Ubuntu

ارائه‌دهنده	آدرس میرور
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir
آروان‌کلود	mirror.arvancloud.ir
شاتل	ubuntu.shatel.ir / mirror.shatel.ir
ایران‌سرور	mirror.iranserver.com
مبین‌هاست	mirror.mobinhost.com
افرانت	mirror.afranet.com
پیشگامان	ubuntu.pishgaman.net
پردیس	mirrors.pardisco.co
شریف	mirrors.sharif.ir
تهران	mirror.ut.ac.ir
میرور رسمی ایران	ir.archive.ubuntu.com

Arch Linux

ارائه‌دهنده	آدرس میرور
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir
آروان‌کلود	mirror.arvancloud.ir
مبین‌هاست	mirror.mobinhost.com
لیارا	liara.ir/mirrors

Fedora

ارائه‌دهنده	آدرس میرور
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir
کرگدن	mirror.kargadan.ir
مبین‌هاست	mirror.mobinhost.com
سازمان فناوری اطلاعات ایران	repo-portal.ito.gov.ir
لیارا	liara.ir/mirrors

Rocky Linux

ارائه‌دهنده	آدرس میرور
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir
کرگدن	mirror.kargadan.ir
سازمان فناوری اطلاعات ایران	repo-portal.ito.gov.ir
لیارا	liara.ir/mirrors

openSUSE

ارائه‌دهنده	آدرس میرور
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir
آروان‌کلود	mirror.arvancloud.ir
کرگدن	mirror.kargadan.ir
لیارا	liara.ir/mirrors

Alpine Linux

ارائه‌دهنده	آدرس میرور
دانشگاه صنعتی اصفهان IUT	repo.iut.ac.ir
آروان‌کلود	mirror.arvancloud.ir
مبین‌هاست	mirror.mobinhost.com
لیارا	liara.ir/mirrors

Linux Mint

آدرس میرور	ارائه‌دهنده
repo.iut.ac.ir	دانشگاه صنعتی اصفهان IUT

Manjaro

آدرس میرور	ارائه‌دهنده
repo.iut.ac.ir	دانشگاه صنعتی اصفهان IUT
mirror.arvancloud.ir	آروان‌کلود
mirror.mobinhost.com	مبین‌هاست
liara.ir/mirrors	لیارا

Kali Linux

آدرس میرور	ارائه‌دهنده
mirror.shatel.ir	شاتل

CentOS

آدرس میرور	ارائه‌دهنده
mirror.arvancloud.ir	آروان‌کلود
mirror.iranserver.com	ایران‌سرور
mirror.afranet.com	افرانت
repo-portal.ito.gov.ir	سازمان فناوری اطلاعات ایران
liara.ir/mirrors	لیارا

FreeBSD

آدرس میرور	ارائه‌دهنده
mirror.mobinhost.com	مبین‌هاست

Raspbian

آدرس میرور	ارائه‌دهنده
repo.iut.ac.ir	دانشگاه صنعتی اصفهان IUT
mirror.mobinhost.com	مبین‌هاست

بخش ششم

تنظیم مخازن داخلی برای توزیع Debian

درDebian ، برای استفاده از یک میرور دیگر، معمولاً کافی است نشانی مخزن (Mirror URL) را تغییر دهید و کدنام نسخه و اجزای مخزن را متناسب با سیستم خود نگه دارید. بااین‌حال، محل تنظیمات مخازن در نسخه‌های مختلف ممکن است متفاوت باشد.

گام اول: شناسایی نسخهDebian

برای مشاهده کدنام نسخه اجرا کنید:

```
. /etc/os-release
echo "$VERSION_CODENAME"
```

برای نمونه، خروجی Debian 13 برابر trixieو خروجی Debian 12 برابر bookwormاست.

گام دوم: شناسایی فایل‌های مخازن

برای مشاهده فایل‌های رایج مخازن اجرا کنید:

```
cat /etc/apt/sources.list
ls -l /etc/apt/sources.list.d/
```

در Debian 13 ممکن است تنظیمات اصلی در فایل زیر قرار داشته باشد:

```
/etc/apt/sources.list.d/debian.sources
```

فایل‌های دارای پسوند . listو . sourcesدر پوشه sources.list.dنیز ممکن است فعال باشند. پیش از تغییر، فایل‌های فعال را بررسی کنید تا مخازن تکراری یا ناسازگار ایجاد نشوند.

گام سوم: پشتیبان‌گیری

برای تهیه نسخه پشتیبان از تنظیمات مخازن اجرا کنید:

```
sudo cp -a /etc/apt/sources.list /etc/apt/sources.list.bak 2>/dev/null || true
sudo cp -a /etc/apt/sources.list.d "/etc/apt/sources.list.d.bak"
```

این دستورات فایل اصلی، در صورت وجود، و پوشه مخازن را پشتیبان‌گیری می‌کنند. اگر قبلاً نسخه پشتیبان دارید، نام متفاوتی انتخاب کنید تا نسخه قبلی بازنویسی نشود.

گام چهارم: تنظیم میرور آروان‌کلود برای نمونه. بقیه میرورها به همین صورت است

برای ویرایش فایل قدیمی مخازن:

```
sudo nano /etc/apt/sources.list
```

نمونه زیر برای Debian 13 با کدنام trixieاست:

```
deb https://mirror.arvancloud.ir/debian trixie main
deb https://mirror.arvancloud.ir/debian trixie-updates main
deb https://mirror.arvancloud.ir/debian-security trixie-security main
```

نکته :پیش از استفاده، دسترس‌پذیری مسیرهای دقیق میرور و پشتیبانی آن از مجموعه‌های موردنیاز را بررسی کنید. اگر میرور امنیتی در نشانی جداگانه ارائه نشده باشد، نباید فرض کنید مسیر بالا حتماً معتبر است.

در صورت نیاز به اجزای دیگر، مانند contrib، non-freeیا non-free-firmware، آن‌ها را متناسب با نیاز سیستم به انتهای خطوط اضافه کنید.

گام پنجم: تنظیم مخازن در قالب جدیدDebian 13

اگر مخازن سیستم در فایل debian.sourcesقرار دارند، آن را ویرایش کنید:

```
sudo nano /etc/apt/sources.list.d/debian.sources
```

نمونه قالب Deb822 برای:Debian 13

```
Types: deb
URIs: https://mirror.arvancloud.ir/debian
Suites: trixie trixie-updates
Components: main
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg

Types: deb
URIs: https://mirror.arvancloud.ir/debian-security
Suites: trixie-security
Components: main
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg
```

این نمونه فقط در صورتی معتبر است که هر دو نشانی واقعاً در میرور ارائه شوند. اگر مخزن امنیتی در دسترس نیست، نشانی رسمی امنیتی Debian را نگه دارید:

```
URIs: https://security.debian.org/debian-security
```

از تعریف هم‌زمان مخازن تکراری در فایل‌های قدیمی و جدید خودداری کنید.

گام ششم: به‌روزرسانی فهرست بسته‌ها

پس از ذخیره تنظیمات اجرا کنید:

```
sudo apt update
```

اگر خطای 404 Not Found مشاهده کردید، نشانی میرور، کدنام نسخه و وجود مسیر موردنظر را بررسی کنید. اگر خطای امضای مخزن مشاهده شد، بررسی صحت مخزن و کلیدهای امضا ضروری است؛ اعتبارسنجی امضا را غیرفعال نکنید.

پس از موفقیت در به‌روزرسانی فهرست بسته‌ها، می‌توانید برای ارتقای بسته‌های نصب‌شده از دستور زیر استفاده کنید:

```
sudo apt upgrade
```

سایر مخازن جایگزین برای دبیان

۲. *شاتل Shatel*)

میرور شاتل قالب کاملی برای نسخه‌های مختلف Debian ارائه می‌دهد . **مثال برای: Debian 13 (Trixie)**

```
deb http://mirror.shatel.ir/debian trixie main contrib non-free non-free-firmware
deb http://mirror.shatel.ir/debian trixie-updates main contrib non-free non-free-firmware
deb http://mirror.shatel.ir/debian trixie-backports main contrib non-free non-free-firmware
deb http://mirror.shatel.ir/debian-security trixie-security main contrib non-free non-free-firm
ware
```

۳. *لیارا Liara*)

لیارا مستندات اختصاصی برای Debian دارد . **مثال برای: Debian 13 (Trixie)**

```
deb https://linux-mirror.liara.ir/repository/debian trixie main non-free-firmware
deb https://linux-mirror.liara.ir/repository/debian trixie-updates main non-free-firmware
deb https://linux-mirror.liara.ir/repository/debian-security trixie-security main non-free-firmw
are
```

۴. *دانشگاه صنعتی اصفهان IUT*)

آدرس پایه repo.iut.ac.ir است و پوشش جامعی برای Debian دارد . مسیر دقیق Debian در این میرور معمولاً به صورت https://repo.iut.ac.ir/debian/ است. قالب خطوط مشابه موارد بالاست:

```
deb https://repo.iut.ac.ir/debian trixie main contrib non-free
```

```
deb https://repo.iut.ac.ir/debian-security trixie-security main contrib non-free
```

۵. ایران‌سرور *IranServer*)

آدرس پایه mirror.iranserver.com است و Debian را پوشش می‌دهد. مسیر Debian در این میرور به صورت https://mirror.iranserver.com/debian/ است . قالب خطوط:

```
deb https://mirror.iranserver.com/debian trixie main contrib non-free
deb https://mirror.iranserver.com/debian-security trixie-security main contrib non-free
```

۶. مبین‌هاست *MobinHost*)

میرور Debian مبین‌هاست در آدرس debian.mobinhost.com (یا زیرمسیر mirror.mobinhost.com/debian/) قرار دارد و هر سه ساعت یک‌بار به‌روزرسانی می‌شود . قالب خطوط:

```
deb http://debian.mobinhost.com/debian trixie main contrib non-free
deb http://debian.mobinhost.com/debian-security trixie-security main contrib non-free
```

۷. پردیس *PardisCo*)

میرور mirrors.pardisco.co در مستندات Debian به عنوان یکی از URI های معتبر ذکر شده است . مسیر Debian در این میرور به صورت https://mirrors.pardisco.co/debian/ است. قالب خطوط:

```
deb https://mirrors.pardisco.co/debian trixie main contrib non-free
deb https://mirrors.pardisco.co/debian-security trixie-security main contrib non-free
```

۸. ابر دانا *Abrha / ParsPack*)

مستندات پارس‌پک که از دامنه repo.abrha.net استفاده می‌کند) دستورالعمل کاملی برای Debian ارائه داده است .

مثال برای Debian 13 (Trixie):

```
deb https://repo.abrha.net/debian trixie main contrib non-free
deb https://repo.abrha.net/debian trixie-updates main contrib non-free
deb https://repo.abrha.net/debian trixie-backports main contrib non-free
deb https://repo.abrha.net/debian-security trixie-security main contrib non-free
```

مخازن داخلی برای توزیع اوبونتو

درUbuntu ، مخازن نرم‌افزاری از طریق APT مدیریت می‌شوند. برای استفاده از یک میرور دیگر، باید نشانی مخزن را تغییر دهید و کدنام نسخه، اجزای مخزن و مخازن به‌روزرسانی و امنیتی را متناسب با نسخه نصب‌شده حفظ کنید.

گام اول: شناسایی نسخهUbuntu

برای مشاهده کدنام نسخه اجرا کنید:

```
lsb_release -cs
```

همچنین می‌توانید از دستور زیر استفاده کنید:

```
. /etc/os-release
echo "$VERSION_CODENAME"
```

برای نمونه، کدنام Ubuntu 24.04 برابر noble، نسخه ۲۲,۰۴ برابر jammyو نسخه ۲۰,۰۴ برابر focalاست.

گام دوم: شناسایی فایل مخازن

ابتدا فایل اصلی را بررسی کنید:

```
cat /etc/apt/sources.list
```

سپس فایل‌های موجود در پوشه مخازن را فهرست کنید:

```
ls -l /etc/apt/sources.list.d/
```

در Ubuntu 24.04 و برخی نسخه‌های جدیدتر، تنظیمات اصلی مخازن ممکن است در فایل زیر قرار داشته باشد:

```
/etc/apt/sources.list.d/ubuntu.sources
```

در این حالت، تغییر فایل / etc/apt/sources.listبه‌تنهایی ممکن است کافی نباشد.

گام سوم: پشتیبان‌گیری

برای پشتیبان‌گیری از فایل اصلی و پوشه مخازن اجرا کنید:

```
sudo cp -a /etc/apt/sources.list /etc/apt/sources.list.bak
sudo cp -a /etc/apt/sources.list.d /etc/apt/sources.list.d.bak
```

اگر فایل / etc/apt/sources.listوجود ندارد، دستور نخست ممکن است خطا بدهد؛ در این حالت، پشتیبان‌گیری از پوشه sources.list.dاهمیت بیشتری دارد.

گام چهارم: ویرایش نشانی مخازن

اگر سیستم از قالب قدیمی استفاده می‌کند، فایل زیر را باز کنید:

```
sudo nano /etc/apt/sources.list
```

اگر مخازن در قالب Deb822 تعریف شده‌اند، فایل مربوطه را ویرایش کنید:

```
sudo nano /etc/apt/sources.list.d/ubuntu.sources
```

در قالب قدیمی، نمونه خطوط برای Ubuntu 24.04 به شکل زیر است:

```
deb https://archive.ubuntu.com/ubuntu noble main restricted universe multiverse
deb https://archive.ubuntu.com/ubuntu noble-updates main restricted universe multiverse
deb https://archive.ubuntu.com/ubuntu noble-backports main restricted universe multiverse
deb https://security.ubuntu.com/ubuntu noble-security main restricted universe multiverse
```

این خطوط نمونه‌ای برای مخازن استاندارد Ubuntu هستند. برای استفاده از یک میرور دیگر، نشانی سرور را با نشانی معتبر میرور موردنظر جایگزین کنید و کدنام و اجزای مخزن را حفظ کنید.

نکته مهم :همه میرورها الزاماً تمام اجزای مخزن یا مخزن امنیتی را ارائه نمی‌کنند. پیش از جایگزینی، پشتیبانی میرور از مسیرهای موردنیاز را بررسی کنید. همچنین، اگر مخازن اختصاصی مانند PPA یا مخازن نرم‌افزارهای شخص ثالث دارید، بدون بررسی آن‌ها را حذف نکنید.

گام پنجم: به‌روزرسانی فهرست بسته‌ها

پس از ذخیره فایل، اجرا کنید:

```
sudo apt update
```

در صورت نبود خطا، فهرست بسته‌ها از نشانی‌های جدید دریافت می‌شود. خطاهای 404 Not Foundمعمولاً به نشانی نادرست، نبود مسیر یا کدنام اشتباه مربوط هستند. خطاهای امضا نیز باید از طریق بررسی اعتبار مخزن و کلیدهای رسمی برطرف شوند؛ اعتبارسنجی امضا را غیرفعال نکنید.

در صورت نیاز به ارتقای بسته‌های نصب‌شده، پس از بررسی خروجی دستور زیر را اجرا کنید:

```
sudo apt upgrade
```

تغییر میرور صرفاً منبع دریافت بسته‌ها را تغییر می‌دهد و به‌خودی‌خود نسخه Ubuntu را ارتقا نمی‌دهد.

سایر مخازن داخلی برای Ubuntu 26.04

در نمونه‌های زیر، مخزن backportsحذف شده است. برای هر میرور، چهار خط زیر را جایگزین تنظیمات مخزن کنید.

۱. دانشگاه صنعتی اصفهان (IUT)

```
deb http://repo.iut.ac.ir/repo/Ubuntu/ resolute main restricted universe multiverse
deb http://repo.iut.ac.ir/repo/Ubuntu/ resolute-updates main restricted universe multiverse
deb http://repo.iut.ac.ir/repo/Ubuntu/ resolute-security main restricted universe multiverse
```

۲. آروان‌کلود

```
deb https://mirror.arvancloud.ir/ubuntu/ resolute main restricted universe multiverse
deb https://mirror.arvancloud.ir/ubuntu/ resolute-updates main restricted universe multiverse
deb https://mirror.arvancloud.ir/ubuntu/ resolute-security main restricted universe multiverse
```

۳. شاتل

```
deb https://ubuntu.shatel.ir/ubuntu/ resolute main restricted universe multiverse
deb https://ubuntu.shatel.ir/ubuntu/ resolute-updates main restricted universe multiverse
deb https://ubuntu.shatel.ir/ubuntu/ resolute-security main restricted universe multiverse
```

۴. ایران‌سرور

```
deb https://mirror.iranserver.com/ubuntu/ resolute main restricted universe multiverse
deb https://mirror.iranserver.com/ubuntu/ resolute-updates main restricted universe multiverse
deb https://mirror.iranserver.com/ubuntu/ resolute-security main restricted universe multiverse
```

۵. مبین‌هاست

```
deb https://mirror.mobinhost.com/ubuntu/ resolute main restricted universe multiverse
deb https://mirror.mobinhost.com/ubuntu/ resolute-updates main restricted universe multiverse
deb https://mirror.mobinhost.com/ubuntu/ resolute-security main restricted universe multiverse
```

۶. افرانت

```
deb http://mirror.afranet.com/ubuntu/ resolute main restricted universe multiverse
deb http://mirror.afranet.com/ubuntu/ resolute-updates main restricted universe multiverse
deb http://mirror.afranet.com/ubuntu/ resolute-security main restricted universe multiverse
```

۷. پیشگامان

```
deb https://ubuntu.pishgaman.net/ubuntu/ resolute main restricted universe multiverse
deb https://ubuntu.pishgaman.net/ubuntu/ resolute-updates main restricted universe multiverse
deb https://ubuntu.pishgaman.net/ubuntu/ resolute-security main restricted universe multiverse
```

۸. پردیس

```
deb https://mirrors.pardisco.co/ubuntu/ resolute main restricted universe multiverse
deb https://mirrors.pardisco.co/ubuntu/ resolute-updates main restricted universe multiverse
deb https://mirrors.pardisco.co/ubuntu/ resolute-security main restricted universe multiverse
```

۹. شریف

```
deb https://mirrors.sharif.ir/ubuntu/ resolute main restricted universe multiverse
deb https://mirrors.sharif.ir/ubuntu/ resolute-updates main restricted universe multiverse
deb https://mirrors.sharif.ir/ubuntu/ resolute-security main restricted universe multiverse
```

۱۰. دانشگاه تهران

```
deb https://mirror.ut.ac.ir/ubuntu/ resolute main restricted universe multiverse
deb https://mirror.ut.ac.ir/ubuntu/ resolute-updates main restricted universe multiverse
deb https://mirror.ut.ac.ir/ubuntu/ resolute-security main restricted universe multiverse
```

۱۱. میرور رسمی ایران

```
deb https://ir.archive.ubuntu.com/ubuntu/ resolute main restricted universe multiverse
deb https://ir.archive.ubuntu.com/ubuntu/ resolute-updates main restricted universe multiverse
deb https://ir.archive.ubuntu.com/ubuntu/ resolute-security main restricted universe multiverse
```

مخازن داخلی برای توزیع آرچ لینوکس

برای Arch Linux ، روش جایگزینی مخزن بسیار ساده است و فقط شامل ویرایش یک فایل و جایگزینی یک خط است. در زیر روش عمومی و سپس آدرس‌های دقیق میرور (مخزن) برای هر ارائه‌دهنده آمده است.

گام ۱: پشتیبان‌گیری از فایل فعلی

```
sudo cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak
```

گام ۲: ویرایش فایل میرور

```
sudo nano /etc/pacman.d/mirrorlist
```

گام ۳: جایگزینی محتوا

کل محتوای فایل را حذف کنید (یا ابتدای هر خط # قرار دهید) و **فقط** خط Server مربوط به میرور مورد نظر خود را وارد کنید.

گام ۴: به‌روزرسانی

```
sudo pacman -Syy
```

آدرس دقیق میرور هر ارائه‌دهنده

۱. آروان‌کلود ArvanCloud)

آروان‌کلود به طور رسمی از Arch Linux پشتیبانی می‌کند . این خط را به عنوان تنها میرور وارد کنید :

```
Server = https://mirror.arvancloud.ir/archlinux/$repo/os/$arch
```

۲. مبین‌هاست MobinHost)

مبین‌هاست یک میرور رسمی ثبت‌شده Arch Linux است و پهنای باند بالایی دارد . خط زیر را وارد کنید :

```
Server = https://mirror.mobinhost.com/archlinux/$repo/os/$arch
```

۳. دانشگاه صنعتی اصفهان IUT)

دانشگاه صنعتی اصفهان یک میرور جامع است که Arch Linux را نیز پوشش می‌دهد . خط زیر را وارد کنید :

```
Server = http://repo.iut.ac.ir/repo/archlinux/$repo/os/$arch
```

۴. لیارا (Liara)

مستندات رسمی لیارا نشان می‌دهد که از میرور Arch Linux پشتیبانی می‌کند ، اما **در نتایج جستجو آدرس دقیق خط** Server **برای Arch در لیارا یافت نشد**. پیشنهاد می‌شود به صفحه مستندات میرور لیارا مراجعه کنید یا از الگوی زیر استفاده کرده و آن را تست نمایید (مسیر معمولاً مشابه ubuntu یا debian است

```
Server = https://linux-mirror.liara.ir/repository/archlinux/$repo/os/$arch
```

سایرمخازن

علاوه بر چهار مورد بالا، میرورهای ایرانی زیر نیز در نتایج جستجو برای Arch Linux توصیه شده‌اند و می‌توانید آن‌ها را به عنوان گزینه‌های جایگزین در فایل mirrorlist قرار دهید :

```
Server = https://mirror.0-1.cloud/archlinux/$repo/os/$arch
Server = https://mirror.kernel.ir/archlinux/$repo/os/$arch
Server = https://mirror-linux.runflare.com/archlinux/$repo/os/$arch
```

مخازن داخلی برای توزیع فدورا

برایFedora ، روش جایگزینی مخزن (Mirror) برخلاف Debian/Ubuntu فقط شامل تغییر یک فایل نیست، بلکه باید فایل‌های مخزن مختلف مثل fedora.repo وfedora-updates.repo را ویرایش یا جایگزین کنید. در ادامه ابتدا روش عمومی و سپس دستورالعمل اختصاصی برای هر ارائه‌دهنده‌ای که مستندات دقیق آن در نتایج جستجو موجود بود، ارائه می‌شود.

گام ۱: پشتیبان‌گیری از فایل‌های فعلی

```
sudo cp /etc/yum.repos.d/fedora.repo /etc/yum.repos.d/fedora.repo.bak
sudo cp /etc/yum.repos.d/fedora-updates.repo /etc/yum.repos.d/fedora-updates.repo.bak
```

گام ۲: غیرفعال کردن metalink در صورت وجود

برخی میرورها نیاز دارند که خط metalinkغیرفعال شود تا baseurlمحلی استفاده شود. می‌توانید این کار را به صورت دستی در فایل انجام دهید یا از دستور sedاستفاده کنید.

گام ۳: جایگزینی محتوای فایل‌های مخزن

با یک ویرایشگر (مثل nanoفایل‌های / etc/yum.repos.d/fedora.repoو / etc/yum.repos.d/fedora-updates.repoرا باز کرده و محتوای آن‌ها را با تنظیمات میرور مورد نظر جایگزین کنید.

گام ۴: پاکسازی و به‌روزرسانی

```
sudo dnf  clean all
sudo dnf  makecache
```

سایر مخازن برای فدورا

۱. کرگدنKargadan)

کرگدن مستندات کامل و رسمی برای Fedora ارائه داده است که شامل تنظیمات آماده برای نسخه‌های مختلف مثل Fedora 42و ۴۳ است . طبق این مستندات، فایل / etc/yum.repos.d/fedora.repoرا با محتوای زیر جایگزین کنید این نمونه برای Fedora 42 است

```
[fedora]
name=Fedora $releasever - $basearch
baseurl=https://mirror.kargadan.ir/repository/fedora-group/releases/$releasever/Everything/$basearch/os/
enabled=1
countme=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch
skip_if_unavailable=False
```

```
[updates]
name=Fedora $releasever - $basearch - Updates
baseurl=https://mirror.kargadan.ir/repository/fedora-group/updates/$releasever/Everything/$basearch/
enabled=1
countme=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch
```

نکته مهم :کرگدن همچنین میرور EPEL را پوشش می‌دهد. اگر به EPEL نیاز دارید، می‌توانید بخش [epel]را نیز (طبق مستندات کرگدن) به فایل اضافه کنید .

۲. لیارا(Liara)

لیارا یک صفحه مستندات اختصاصی برای Fedora دارد . مراحل طبق مستندات لیارا:

گام ۱: بکاپ

```
sudo cp /etc/yum.repos.d/fedora.repo /etc/yum.repos.d/fedora.repo.bak
sudo cp /etc/yum.repos.d/fedora-updates.repo /etc/yum.repos.d/fedora-updates.repo.bak
```

۳. دانشگاه صنعتی اصفهان(IUT)

IUTیک میرور جامع است که Fedora را پوشش می‌دهد . بر اساس ساختار مسیر موجود در نتایج جستجو مثلاً repo.iut.ac.ir/repo/fedora/linux/releases/44/Everything/...، می‌توانید از الگوی baseurlزیر استفاده کنید:

```
[fedora]
name=Fedora $releasever - $basearch
baseurl=https://repo.iut.ac.ir/repo/fedora/linux/releases/$releasever/Everything/$basearch/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch
```

برای بخش [updates]می‌توانید از مسیر مشابه

```
repo.iut.ac.ir/repo/fedora/linux/updates/$releasever/Everything/$basearch/
```

استفاده کنید، اما این مسیر در نتایج جستجو مستقیماً تأیید نشده است.

۴. مبین‌هاست(MobinHost)

مبین‌هاست در فهرست‌های میرور داخلی به عنوان پوشش‌دهنده Fedora و EPEL ذکر شده است ، اما **آدرس دقیق مسیر Fedoraدر نتایج جستجو یافت نشد**. پیشنهاد می‌شود به وب‌سایت mirror.mobinhost.comمراجعه کرده و مسیر Fedoraرا پیدا کنید، یا از الگوی https://mirror.mobinhost.com/fedora/استفاده کرده و با curlتست کنید.

مخازن داخلی برای توزیع راکی لینوکس

برایRocky Linux ، روش جایگزینی مخزن مشابه Fedora است و باید فایل‌های موجود در /etc/yum.repos.d/مخصوصاً rocky.repoویرایش شوند. در ادامه ابتدا روش عمومی و سپس دستورالعمل اختصاصی برای هر ارائه‌دهنده‌ای که مستندات دقیق آن در نتایج جستجو موجود بود، ارائه می‌شود.

گام ۱: پشتیبان‌گیری از فایل‌های فعلی

```
sudo cp /etc/yum.repos.d/rocky.repo /etc/yum.repos.d/rocky.repo.bak
```

گام ۲: جایگزینی محتوای فایل مخزن

با یک ویرایشگر مثل nano فایل /etc/yum.repos.d/rocky.repoرا باز کرده و محتوای آن را با تنظیمات میرور مورد نظر جایگزین کنید.

گام ۳: پاکسازی و به‌روزرسانی

```
sudo dnf  clean all
sudo dnf  update
```

سایرمخازن برای راکی لینوکس

۱. لیارا(Liara)

لیارا مستندات رسمی و کاملی برای Rocky Linux ارائه داده است که شامل تنظیمات آماده برای مخازن baseos، appstreamو crbاست . طبق این مستندات، فایل /etc/yum.repos.d/rocky.repoرا با محتوای زیر جایگزین کنید:

```
[baseos]
name=Rocky Linux $releasever - BaseOS
baseurl=https://linux-mirror.liara.ir/repository/rocky/$contentdir/$releasever/BaseOS/$basearch/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9
metadata_expire=6h

[appstream]
name=Rocky Linux $releasever - AppStream
baseurl=https://linux-mirror.liara.ir/repository/rocky/$contentdir/$releasever/AppStream/$basearch/os
/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9
metadata_expire=6h
```

```
[crb]
name=Rocky Linux $releasever - CRB
baseurl=https://linux-mirror.liara.ir/repository/rocky/$contentdir/$releasever/CRB/$basearch/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9
metadata_expire=6h
```

نکته :در نسخه‌های Rocky Linux 9 و ۱۰، مخزن crb مخففCode Ready Builder وجود دارد و برای نصب برخی بسته‌های اضافی مثل EPEL ممکن است نیاز به فعال بودن آن باشد . لیارا این مخزن را نیز در تنظیمات خود گنجانده است.

۲. دانشگاه صنعتی اصفهان(IUT)

IUTیک میرور جامع است و مسیر Rocky Linux در آن به صورت repo.iut.ac.ir/repo/rocky-linux/است . ساختار پوشه‌ها نشان می‌دهد که مسیرها شامل نسخه (مثل 10یا 8.10و سپس BaseOSیا AppStreamو معماری x86_64) یا riscv64)و در نهایت osاست. بنابراین، می‌توانید از الگوی baseurlزیر استفاده کنید (به جای 9، نسخه اصلی خود را قرار دهید

```
[baseos]
name=Rocky Linux $releasever - BaseOS
baseurl=https://repo.iut.ac.ir/repo/rocky-linux/$releasever/BaseOS/$basearch/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9

[appstream]
name=Rocky Linux $releasever - AppStream
baseurl=https://repo.iut.ac.ir/repo/rocky-linux/$releasever/AppStream/$basearch/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-Rocky-9
```

مخازن داخلی برای توزیع مینت

برایLinux Mint ، روش جایگزینی مخزن کمی با اوبونتو متفاوت است، زیرا مینت به **دو مخزن همزمان** نیاز دارد: مخزن بسته‌های خود مینت (Mint packages) و مخزن پایه اوبونتو .(Ubuntu base) بر اساس نتایج جستجو، در حال حاضر **دو ارائه‌دهنده اصلی** برای مینت در ایران قابل استفاده هستند :**کرگدن (Kargadan)** با مستندات کامل و **دانشگاه صنعتی اصفهان (IUT)** با مسیرهای موجود.

درLinux Mint ، فایل اصلی مخازن / etc/apt/sources.list.d/official-package-repositories.list است نه:

/etc/apt/sources.list

گام ۱: پشتیبان‌گیری

```
sudo cp /etc/apt/sources.list.d/official-package-repositories.list /etc/apt/sources.list.d/official-package-repositories.list.bak
```

گام ۲: ویرایش فایل

```
sudo nano /etc/apt/sources.list.d/official-package-repositories.list
```

گام ۳: جایگزینی مخزن جدید

مخزن Kargadan برای مینت

Kargadanیک آینه apt مخصوص Linux Mint ارائه می‌دهد که شامل آرشیو بسته‌های Mint و آینه پایه Ubuntu که Mint به آن وابسته است می‌شود .

گام ۱: پشتیبان‌گیری از فایل پیکربندی اصلی

```
sudo cp /etc/apt/sources.list.d/official-package-repositories.list /etc/apt/sources.list.d/official-package-repositories.list.bak
```

گام ۲: ویرایش فایل پیکربندی

```
sudo nano /etc/apt/sources.list.d/official-package-repositories.list
```

گام ۳: جایگزینی با محتوای جدید برای "Linux Mint 22 "Wilma به عنوان مثال

کل فایل را پاک کنید و محتوای زیر را جای‌گذاری نمایید این پیکربندی از مستندات رسمی Kargadan است

```
# Linux Mint 22 "Wilma"
deb [trusted=yes] https://mirror.kargadan.ir/repository/linuxmint-proxy wilma main upstream import backport

# Ubuntu base (24.04)
```

```
deb [trusted=yes] https://mirror.kargadan.ir/repository/ubuntu-proxy noble main restricted universe multiverse
deb [trusted=yes] https://mirror.kargadan.ir/repository/ubuntu-proxy noble-updates main restricted universe multiverse
deb [trusted=yes] https://mirror.kargadan.ir/repository/ubuntu-proxy noble-backports main restricted universe multiverse
deb [trusted=yes] https://mirror.kargadan.ir/repository/ubuntu-security-proxy noble-security main restricted universe multiverse
```

توجه :اگر از نسخه دیگری استفاده می‌کنید مانند"Mint 21.3 "Virginia ط، باید wilmaرا به virginiaو nobleرا به jammyتغییر دهید. Kargadan برای نسخه‌های مختلف دستورالعمل‌های متناظر ارائه می‌دهد .

گام ۴: به‌روزرسانی کش

```
sudo apt update
```

مخازن داخلی برای توزیع Alpine Linux

در توزیع Alpine Linuxبرای تغییر مخزن بسته‌ها (Repositoryباید فایل زیر را ویرایش کنید:

```
/etc/apk/repositories
```

روش کلی تغییر مخزن

ابتدا نسخه Alpineرا بررسی کنید:

```
cat /etc/alpine-release
```

سپس از فایل مخازن نسخه پشتیبان بگیرید:

```
cp /etc/apk/repositories /etc/apk/repositories.backup
```

فایل را باز کنید:

```
vi /etc/apk/repositories
```

محتوای پیش‌فرض معمولاً مشابه نمونه زیر است. در این مثال نسخه Alpineبرابر 3.24است:

```
https://dl-cdn.alpinelinux.org/alpine/v3.24/main
https://dl-cdn.alpinelinux.org/alpine/v3.24/community
```

آدرس مخزن را با میرور موردنظر جایگزین کنید و شماره نسخه و شاخه‌های mainو communityرا حفظ کنید.

سایر مخازن الپاین

با فرض اینکه نسخه شما ۳,۲۴ است (لطفاً بر اساس cat /etc/alpine-releaseواقعی خود تنظیم کنید)، محتوای فایل / etc/apk/repositoriesرا با موارد زیر جایگزین کنید:

```
https://linux-mirror.liara.ir/repository/alpine/v3.24/main
https://linux-mirror.liara.ir/repository/alpine/v3.24/community
```

سپس apk updateرا اجرا کنید.

تکمیل سایر میرورهای Alpine

در آموزش قبلی شما، میرورهای IUT ، ArvanCloudو MobinHost ذکر شده بودند. در نتایج جستجو، **Kargadan**نیز میرور Alpine ارائه می‌دهد که آدرس آن به شرح زیر است :

```
https://mirror.kargadan.ir/repository/alpine-proxy/v3.24/main
https://mirror.kargadan.ir/repository/alpine-proxy/v3.24/community
```

اگر میرورهای بالا با مشکل مواجه شدند، می‌توانید این گزینه را نیز امتحان کنید.

به‌روزرسانی مخازن

پس از ذخیره فایل، دستور زیر را اجرا کنید:

```
apk update
```

برای به‌روزرسانی بسته‌های نصب‌شده:

```
apk upgrade
```

در صورت بروز مشکل می‌توانید تنظیمات قبلی را بازگردانید:

```
cp /etc/apk/repositories.backup /etc/apk/repositories
apk update
```

مخازن داخلی برای توزیع Manjaro Linux

توزیع Manjaro Linuxاز مدیر بسته Pacmanاستفاده می‌کند. برای افزایش سرعت دریافت بسته‌ها و به‌روزرسانی سیستم، می‌توان از میرورهای داخلی ایران استفاده کرد.

۱. فهرست میرورهای Manjaro

میرورهای زیر برای توزیع Manjaroمعرفی شده‌اند. مسیر مخزن هر ارائه‌دهنده را نیز در جدول آورده‌ام.

ارائه‌دهنده	آدرس مخزن Manjaro
دانشگاه صنعتی اصفهان (IUT)	https://repo.iut.ac.ir/repo/manjaro/
آروان‌کلود	https://mirror.arvancloud.ir/manjaro/
مبین‌هاست	https://mirror.mobinhost.com/manjaro/
لیارا	https://liara.ir/mirrors/manjaro/
Kargadan	https://mirror.kargadan.ir/repository/manjaro-proxy/

مسیرهای IUTو مبین‌هاست در فهرست رسمی وضعیت میرورهای Manjaroقابل مشاهده‌اند. مسیرهای آروان‌کلود و لیارا در این بررسی به‌صورت میرور رسمی Manjaroتأیید نشدند؛ پیش از استفاده باید آزمایش شوند.

ساختار آدرس مخازن

Manjaroسه شاخه اصلی دارد:

- Stable : نسخه پایدار و مناسب استفاده روزمره
- Testing : نسخه‌های در حال آزمایش
- Unstableس : بسته‌های جدیدتر و ناپایدار

ساختار کلی آدرس مخزن به شکل زیر است:

```
https://SERVER/PATH/BRANCH/$repo/$arch
```

در این ساختار:

- SERVER/PATH : آدرس میرور
- BRANCH : نام شاخه، مانندstable
- $repo : نام مخزن بسته، مانندcore یاextra
- $arch : معماری سیستم، معمولاً x86_64

تنظیم مخازن در Pacman

برای استفاده از هر میرور، فایل زیر را ویرایش کنید:

```
sudo nano /etc/pacman.d/mirrorlist
```

محتوای فایل را می‌توانید مطابق میرور موردنظر تنظیم کنید.

دانشگاه صنعتی اصفهان (IUT):

```
Server = https://repo.iut.ac.ir/repo/manjaro/stable/$repo/$arch
```

آروان‌کلود:

```
Server = https://mirror.arvancloud.ir/manjaro/stable/$repo/$arch
```

مبین‌هاست:

```
Server = https://mirror.mobinhost.com/manjaro/stable/$repo/$arch
```

لیارا:

```
Server = https://liara.ir/mirrors/manjaro/stable/$repo/$arch
```

Kargadan

```
Server = https://mirror.kargadan.ir/repository/manjaro-proxy/stable/$repo/$arch
```

برای استفاده از شاخه‌های دیگر، کافی است stableرا با testingیا unstableجایگزین کنید؛ البته باید پشتیبانی آن شاخه توسط میرور تأیید شده باشد.

بررسی دسترسی به مخازن

پیش از استفاده، دسترسی به فایل پایگاه داده مخزن coreرا بررسی کنید:

```
curl -I https://repo.iut.ac.ir/repo/manjaro/stable/core/x86_64/core.db

curl -I https://mirror.arvancloud.ir/manjaro/stable/core/x86_64/core.db

curl -I https://mirror.mobinhost.com/manjaro/stable/core/x86_64/core.db

curl -I https://liara.ir/mirrors/manjaro/stable/core/x86_64/core.db

curl -I https://mirror.kargadan.ir/repository/manjaro-proxy/stable/core/x86_64/core.db
```

پاسخ HTTP 200نشان می‌دهد فایل از مسیر موردنظر قابل دریافت است. البته برای اطمینان از عملکرد کامل میرور، باید دریافت بسته‌ها، همگام‌سازی مخزن و اعتبار امضای بسته‌ها نیز بررسی شود.

به‌روزرسانی سیستم

پس از تنظیم میرور، برای همگام‌سازی پایگاه داده و ارتقای بسته‌ها اجرا کنید:

```
sudo pacman -Syyu
```

مخازن داخلی برای توزیع – Kali Linux

آدرس مخزن:

```
https://mirror.kargadan.ir/repository/kali-proxy/
```

فایل تنظیمات:

```
sudo nano /etc/apt/sources.list
```

محتوای پیشنهادی برای شاخه :Rolling

```
deb https://mirror.kargadan.ir/repository/kali-proxy kali-rolling main contrib non-free non-free-firmware
```

برای به‌روزرسانی:

```
sudo apt update
sudo apt full-upgrade
```

برای بررسی دسترسی:

```
curl -I https://mirror.kargadan.ir/repository/kali-proxy/dists/kali-rolling/Release
```

مخازن داخلی برای توزیعParrot Security OS

آدرس مخزن:

```
https://mirror.kargadan.ir/repository/parrot-proxy/
```

فایل تنظیمات:

```
sudo nano /etc/apt/sources.list.d/parrot.list
```

برای Parrot 6با نام رمز :lory

```
deb https://mirror.kargadan.ir/repository/parrot-proxy lory main contrib non-free non-free-firmware
deb https://mirror.kargadan.ir/repository/parrot-proxy lory-security main contrib non-free non-free-firmware
deb https://mirror.kargadan.ir/repository/parrot-proxy lory-backports main contrib non-free non-free-firmware
```

سپس:

```
sudo apt update
sudo apt full-upgrade
```

بررسی مخزن:

```
curl -I https://mirror.kargadan.ir/repository/parrot-proxy/dists/lory/Release
```

مخازن داخلی برای توزیع CentOS

CentOSاز مدیر بسته‌های DNFو YUMاستفاده می‌کند. مسیر مخازن آن با توزیع‌هایی مانند Kaliو Parrot متفاوت است.

ارائه‌دهنده	آدرس
آروان‌کلود	https://mirror.arvancloud.ir/
ایران‌سرور	https://mirror.iranserver.com/
افرانت	http://mirror.afranet.com/
سازمان فناوری اطلاعات ایران	https://repo-portal.ito.gov.ir/
لیارا	https://liara.ir/mirrors

این ارائه‌دهندگان در فهرست میرورهای ایرانی برای توزیع‌های مبتنی بر RPMمعرفی شده‌اند؛ اما دامنه‌های بالا به‌تنهایی مسیر نهایی مخزن CentOSنیستند.

برای CentOS Streamمعمولاً مخازن در مسیرهایی شامل نسخه و معماری سیستم قرار می‌گیرند. بنابراین نباید بدون بررسی، مسیری مانند / centos/را به همه این دامنه‌ها اضافه کرد.

برای مشاهده نسخه سیستم:

```
cat /etc/centos-release
```

و برای بررسی مخازن فعال:

```
dnf repolist
```

در صورت دریافت فایل تنظیمات اختصاصی هر میرور، باید نشانی آن در فایل‌های .repoزیر قرار بگیرد:

```
/etc/yum.repos.d/
```

مخازن داخلی برای توزیعFreeBSD

مبین‌هاست برای FreeBSDمسیر زیر را معرفی کرده است:

مسیر اعلام‌شده ارائه‌دهنده

```
https://mirror.mobinhost.com/FreeBSD/
```

این مسیر را می‌توان برای دسترسی به آرشیو FreeBSDبررسی کرد.

مبین هاست

برای بررسی دسترسی:

```
curl -I https://mirror.mobinhost.com/FreeBSD/
```

نکته مهم :در FreeBSD، دریافت بسته‌های نصب‌شدنی از طریق pkgبا دریافت فایل‌های آرشیوی و سورس تفاوت دارد. مستندات رسمی FreeBSDاعلام می‌کند که پروژه، میرور عمومی برای مخازن باینری بسته‌ها ارائه نمی‌کند؛ بنابراین نباید فرض کرد که این مسیر الزاماً یک مخزن pkgقابل استفاده است.

مخازن داخلی برای توزیع– Raspbian

Raspbianتوزیعی مبتنی بر Debianبرای Raspberry Piاست و از APTاستفاده می‌کند.

ارائه‌دهنده	آدرس مخزن
دانشگاه صنعتی اصفهان (IUT)	http://repo.iut.ac.ir/repo/raspbian/raspbian/
مبین‌هاست	https://mirror.mobinhost.com/raspbian/

هر دو مسیر در منابع مربوط به میرورهای Raspbianمعرفی شده‌اند.

تنظیم میرور دانشگاه صنعتی اصفهان

فایل مخازن را باز کنید:

```
sudo nano /etc/apt/sources.list
```

خط مخزن:

```
deb http://repo.iut.ac.ir/repo/raspbian/raspbian/ bookworm main contrib non-free non-free-firmware
```

تنظیم میرور مبین‌هاست

```
deb https://mirror.mobinhost.com/raspbian/ bookworm main contrib non-free non-free-firmware
```

در این مثال bookwormصرفاً نمونه‌ای برای نسخه‌های مبتنی بر Debian 12است. نام رمز باید با نسخه نصب‌شده روی Raspberry Piمطابقت داشته باشد.

برای بررسی نسخه:

```
cat /etc/os-release
```

برای به‌روزرسانی:

```
sudo apt update
sudo apt upgrade
```

مخازن داخلی برای توزیع Red Hat Enterprise Linux (UBI)

توزیع‌های مبتنی بر Red Hatاز مدیر بسته‌های DNFو YUMاستفاده می‌کنند. میرور Kargadanامکان دسترسی به مخازن عمومی Red Hat Universal Base Image (UBI)را فراهم می‌کند.

این میرور از محتوای عمومی UBIنسخه‌های 8و 9پشتیبانی می‌کند؛ اما مخازن اختصاصی و نیازمند اشتراک Red Hatرا ارائه نمی‌دهد.

۱. آدرس مخازن Kargadan

نوع مخزن	آدرس
مخزن اصلی Red Hat UBI	https://mirror.kargadan.ir/repository/rhel-group/
آرشیو UBI	https://mirror.kargadan.ir/repository/rhel-ubi-proxy/
مخزن EPEL	https://mirror.kargadan.ir/repository/epel-proxy/

- rhel-group: نقطه دسترسی یکپارچه برای مخازن عمومی UBI
- rhel-ubi-proxy: مخزن محتوای عمومی UBIها
- epel-proxy: مخزن بسته‌های اضافی EPEL

ابتدا فایل تنظیمات مخازن را ایجاد یا ویرایش کنید:

```
sudo nano /etc/yum.repos.d/redhat.repo
```

محتوای زیر را در فایل قرار دهید:

```
# Red Hat UBI 9 - BaseOS

[ubi-9-baseos]
name=Red Hat UBI 9 - BaseOS
baseurl=https://mirror.kargadan.ir/repository/rhel-group/ubi9/9/$basearch/baseos/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

# Red Hat UBI 9 - AppStream

[ubi-9-appstream]
name=Red Hat UBI 9 - AppStream
baseurl=https://mirror.kargadan.ir/repository/rhel-group/ubi9/9/$basearch/appstream/os/
enabled=1
gpgcheck=1
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

در این تنظیمات:

- BaseOS: شامل بسته‌های اصلی سیستم‌عامل است.
- AppStream: شامل برنامه‌ها، ابزارها و محیط‌های اجرایی اضافی است.
- basearch$: معماری سیستم را مشخص می‌کند.
- gpgcheck=1: اعتبار امضای بسته‌ها را بررسی می‌کند.

تنظیم مخازن Red Hat UBI 8

برای نسخه UBI 8 از تنظیمات زیر استفاده کنید:

```
# Red Hat UBI 8 - BaseOS

[ubi-8-baseos]
name=Red Hat UBI 8 - BaseOS
baseurl=https://mirror.kargadan.ir/repository/rhel-group/ubi8/8/$basearch/baseos/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

# Red Hat UBI 8 - AppStream

[ubi-8-appstream]
name=Red Hat UBI 8 - AppStream
baseurl=https://mirror.kargadan.ir/repository/rhel-group/ubi8/8/$basearch/appstream/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

فعال‌سازی مخزن EPEL

مخزن EPEL مجموعه‌ای از بسته‌های اضافی برای توزیع‌های Enterprise Linux است.

برای فعال‌سازی آن در UBI 9، این بخش را به فایل تنظیمات اضافه کنید:

```
[epel]
name=Extra Packages for Enterprise Linux 9
baseurl=https://mirror.kargadan.ir/repository/rhel-group/9/Everything/$basearch/
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-9
```

در تنظیمات ارائه‌شده، enabled=0است؛ بنابراین EPELبه‌صورت پیش‌فرض غیرفعال می‌ماند. برای فعال‌سازی، مقدار آن را به 1تغییر دهید.

توجه داشته باشید که کلید EPELباید روی سیستم نصب شده باشد.

بررسی دسترسی به مخازن

برای بررسی مخزن BaseOSدر UBI 9:

```
curl -I https://mirror.kargadan.ir/repository/rhel-group/ubi9/9/x86_64/baseos/os/repodata/repomd.xml
```

برای بررسی AppStream:

```
curl -I https://mirror.kargadan.ir/repository/rhel-group/ubi9/9/x86_64/appstream/os/repodata/repomd.xml
```

برای بررسی UBI 8:

```
curl -I https://mirror.kargadan.ir/repository/rhel-group/ubi8/8/x86_64/baseos/os/repodata/repomd.xml
```

پاسخ HTTP 200نشان می‌دهد فایل متادیتای مخزن قابل دریافت است.

۷. به‌روزرسانی و نصب بسته‌ها

پس از ذخیره تنظیمات، کش مخازن را پاک کنید:

```
sudo dnf clean all
```

سپس فهرست بسته‌ها را به‌روزرسانی کنید:

```
sudo dnf makecache
```

برای مشاهده مخازن فعال:

```
sudo dnf repolist
```

برای به‌روزرسانی سیستم:

```
sudo dnf update
```

برای نصب یک بسته:

```
sudo dnf install PACKAGE_NAME
```

بخش هفتم

جایگزینی مخازن بسته‌های نرم‌افزاری Package Managers)

در این بخش، روش تنظیم مخازن داخلی برای زبان‌ها و ابزارهای برنامه‌نویسی مختلف ارائه می‌شود. هدف، دور زدن تحریم‌ها و افزایش سرعت دانلود بسته‌هاست.

pip / PyPI Python)

برایPyPI ، گزینه‌های متعددی وجود دارد. **لیارا (Liara)** و **رانفلر (Runflare)** معروف‌ترین و پایدارترین گزینه‌ها هستند. **کرگدن (Kargadan)** نیز یک رجیستری گروه‌بندی‌شده ارائه می‌دهد که درخواست‌ها را از چند منبع تأمین می‌کند .

گزینه ۱: لیارا — (Liara) پیشنهادی

لیارا یک میرور اختصاصی و پایدار برای PyPI دارد.

تنظیم دائمی (ویندوز):

```
setx PIP_INDEX_URL "https://package-mirror.liara.ir/repository/pypi/simple"
```

تنظیم دائمی (لینوکس/مک):

```
export PIP_INDEX_URL=https://package-mirror.liara.ir/repository/pypi/simple
```

نصب یک‌باره (بدون تنظیم دائمی):

```
pip install <package_name> -i https://package-mirror.liara.ir/repository/pypi/simple
```

گزینه ۲: رانفلر Runflare)

رانفلر نیز میرور PyPI را ارائه می‌دهد .

تنظیم دائمی:

```
python -m pip config set global.index-url https://mirror-pypi.runflare.com/simple
python -m pip config set global.timeout 60
```

گزینه ۳: کرگدن Kargadan)

کرگدن یک رجیستری گروه‌بندی‌شده (Group) ارائه می‌دهد که درخواست‌ها را از منابع مختلف تأمین می‌کند .

آدرس رجیستری https://mirror.kargadan.ir/repository/pypi-group/simple :

(توجه: این آدرس بر اساس الگوی مشابه npm و yarn در سایت کرگدن است و در نتایج جستجو مستقیماً ذکر نشده است؛ در صورت استفاده، صحت آن را با curl بررسی کنید

(npm JavaScript / Node.js .

برای npm ، گزینه‌های متعددی وجود دارد که هر کدام ویژگی‌های خاص خود را دارند.

گزینه ۱: کرگدن — (Kargadan) جامع‌ترین

کرگدن یک رجیستری گروه‌بندی‌شده برای npm ارائه می‌دهد که با ابزارهای npm ، pnpm، Bun و Yarn سازگار است .

تنظیم دائمی: (npm)

```
npm config set registry https://mirror.kargadan.ir/repository/npm-group/
```

تنظیم برای یک پروژه خاص فایل .: npmrc

```
registry=https://mirror.kargadan.ir/repository/npm-group/
```

نصب یک‌باره:

```
npm install <package_name> --registry https://mirror.kargadan.ir/repository/npm-group/
```

گزینه ۲: رانفلر (Runflare)

رانفلر نیز رجیستری npm را ارائه می‌دهد .

تنظیم دائمی:

```
npm config set registry https://mirror-npm.runflare.com
npm config set strict-ssl false
```

نصب یک‌باره:

```
npm install --registry="https://mirror-npm.runflare.com" <package_name>
```

گزینه ۳: پارس‌پک (ParsPack)

پارس‌پک نیز یک میرور npm ارائه می‌دهد .

تنظیم دائمی:

```
npm config --global set registry https://mirror.abrha.net/repository/npm/
```

Yarn

Yarnاز رجیستری‌های سازگار باnpm استفاده می‌کند، بنابراین می‌توانید از همان میرورهایnpm استفاده کنید. اما تنظیمات آن بسته به نسخه Classic یا Berry متفاوت است.

گزینه ۱: کرگدن(Kargadan)

کرگدن راهنمای اختصاصی برای Yarn 1 (Classic) و Yarn 2+ (Berry) ارائه داده است .

Yarn 1 (Classic):

```
yarn config set registry https://mirror.kargadan.ir/repository/yarn-group/
```

Yarn 2+ (Berry):

```
yarn config set npmRegistryServer https://mirror.kargadan.ir/repository/yarn-group/
```

گزینه ۲: رانفلر(Runflare)

رانفلر نیز از تنظیم رجیستری برای Yarn پشتیبانی می‌کند .

تنظیم:

```
yarn config set npmRegistryServer https://mirror-npm.runflare.com
```

Gradle / Maven . اندروید و جاوا

برای Gradle وMaven ، **مایکت(Myket)** ، **کرگدن (Kargadan)** و **جمکو (Jamko)** گزینه‌های اصلی هستند.

گزینه ۱: مایکت — (Myket) اختصاصی اندروید

مایکت یک میرور پروکسی برای کتابخانه‌های اندروید (Maven Central, Google Maven, JitPack) ارائه می‌دهد .

تنظیم سراسری (پیشنهادی):
یک فایل `init.gradle.kts` در مسیر `~/.gradle/`(لینوکس/مک) یا `C:\Users\<YourUsername>\.gradle\`(ویندوز) بسازید و کد زیر را در آن قرار دهید :

```
import org.gradle.api.artifacts.repositories.MavenArtifactRepository
import org.gradle.api.artifacts.dsl.RepositoryHandler

fun RepositoryHandler.enableMirror() {
    all {
        if (this is MavenArtifactRepository) {
            val originalUrl = this.url.toString().removeSuffix("/")
```

```
            urlMappings[originalUrl]?.let { mirrorUrl ->
                setUrl(mirrorUrl)
            }
        }
    }
}

val urlMappings = mapOf(
    "https://repo.maven.apache.org/maven2" to "https://maven.myket.ir",
    "https://dl.google.com/dl/android/maven2" to "https://maven.myket.ir",
    "https://plugins.gradle.org/m2" to "https://maven.myket.ir/"
)

gradle.allprojects {
    buildscript {
        repositories.enableMirror()
    }
    repositories.enableMirror()
}
```

گزینه ۲: کرگدن(Kargadan)

کرگدن یک رجیستری گروه‌بندی‌شده برای Maven و Gradle ارائه می‌دهد .

تنظیم در `settings.gradle` **یا** `build.gradle`**:**

```
repositories {
    maven { url 'https://mirror.kargadan.ir/repository/gradle-group/' }
    mavenCentral()
}
```

تنظیم در `settings.gradle` **(برای پلاگین‌ها):**

```
pluginManagement {
    repositories {
        maven { url 'https://mirror.kargadan.ir/repository/gradle-group/' }
        gradlePluginPortal()
    }
}
```

جمکو (Jamko)

جمکو یک میرور ایرانی برای Gradle ارائه می‌دهد که در لیست مخازن پیشنهادی برای توسعه‌دهندگان اندروید در ایران ذکر شده است .

آدرس https://gradle.jamko.ir :

تنظیم در settings.gradle.kts :

```
pluginManagement {
    repositories {
        maven { url = uri("https://gradle.jamko.ir") }
        // سایر مخازن...
    }
}
```

گزینه ۴: سایر میرورهای Gradle

در لیست میرورهای پیشنهادی برای اندروید، موارد زیر نیز ذکر شده‌اند :

```
https://maven.devneeds.ir
https://gradle.iranrepo.ir
https://en-mirror.ir
https://archive.ito.gov.ir/gradle/maven-plugin/
```

سایر بسته‌های نرم‌افزاری NuGet ، Composer، Go

کرگدن (Kargadan) علاوه بر موارد بالا، رجیستری‌های زیر را نیز پوشش می‌دهد :

ابزار	دستور تنظیم (از مستندات کرگدن)
Go	go env -w GOPROXY=https://mirror.kargadan.ir/repository/go-group/,direct
NET / NuGet.	dotnet nuget add source https://mirror.kargadan.ir/repository/nuget-group/index.json -n kargadan
PHP / Composer	composer config -g repos.packagist composer /https://mirror.kargadan.ir/repository/composer-jamko

📌جدول خلاصه

ابزار / زبان	میرور پیشنهادی	آدرس (نمونه)
(pip (Python	لیارا	https://package-mirror.liara.ir/repository/pypi/simple
(pip (Python	رانفلر	https://mirror-pypi.runflare.com/simple
npm	کرگدن	https://mirror.kargadan.ir/repository/npm-group/
npm	رانفلر	https://mirror-npm.runflare.com
npm	پارس‌پک	https://mirror.abrha.net/repository/npm/
(Yarn (Classic	کرگدن	https://mirror.kargadan.ir/repository/yarn-group/
(Yarn (Berry	کرگدن	https://mirror.kargadan.ir/repository/yarn-group/
Gradle / Maven	مایکت	https://maven.myket.ir
Gradle / Maven	کرگدن	https://mirror.kargadan.ir/repository/gradle-group/
Gradle / Maven	جمکو	https://gradle.jamko.ir

PHP / Composer.

برای مدیریت وابستگی‌های PHP ، دو گزینه اصلی داخلی وجود دارد که هر دو مستندات رسمی دارند.

گزینه ۱: رانفلر Runflare)

رانفلر یک میرور اختصاصی برای Composer ارائه می‌دهد که با یک دستور ساده تنظیم می‌شود .

تنظیم سراسری (برای همه پروژه‌ها):

```
composer config --global repo.packagist composer https://mirror-composer.runflare.com
```

تنظیم فقط برای پروژه فعلی بدون --global:

```
composer config repo.packagist composer https://mirror-composer.runflare.com
```

نکته :رانفلر برای هر IP سقف ۵۰۰ درخواست رایگان در نظر گرفته است . برای پروژه‌های بزرگ‌تر باید بسته افزایش سقف تهیه کنید.

گزینه ۲: پارس‌پک (ParsPack)

پارس‌پک نیز یک میرور Composer با مستندات کامل ارائه می‌دهد .

تنظیم سراسری:

```
composer config --global repo.packagist composer https://mirror.abrha.net/repository/composer/
```

تنظیم برای پروژه فعلی:

```
composer config repo.packagist composer https://mirror.abrha.net/repository/composer/
```

بررسی وضعیت:

```
composer config --global --list | grep packagist
```

حذف میرور (بازگشت به پیش‌فرض):

```
composer config --global --unset repos.packagist
```

Go Modules)

گزینه ۱: رانفلر Runflare)

رانفلر میرور Go را با تنظیم متغیر محیطی GOPROXY ارائه می‌دهد .

تنظیم دائمی:

```
go env -w GOPROXY=https://mirror-go.runflare.com
```

تنظیم موقت (فقط برای یک بار اجرا):

```
GOPROXY=https://mirror-go.runflare.com go mod download
```

گزینه ۲: کرگدن Kargadan)

کرگدن نیز یک رجیستری گروه‌بندی‌شده برای Go ارائه می‌دهد.

تنظیم دائمی:

```
go env -w GOPROXY=https://mirror.kargadan.ir/repository/go-group/,direct
```

.NET / NuGet

گزینه ۱: لیارا Liara)

لیارا یک میرور اختصاصی NuGet دارد که در انجمن‌های برنامه‌نویسی ایران به آن اشاره شده است .

آدرس مستندات: https://docs.liara.ir/mirrors/nuget/

روش تنظیم در Visual Studio:

۱. به Tools > Options > NuGet Package Manager > Package Sources بروید.
۲. روی +کلیک کنید و آدرس میرور لیارا را وارد کنید.
۳. آن را به عنوان منبع فعال انتخاب کنید.

روش تنظیم (خط فرمان):

```
dotnet nuget add source https://nuget-mirror.liara.ir/v3/index.json -n Liara
```

توجه: آدرس دقیق nuget-mirror.liara.ir *بر اساس الگوی سایر میرورهای لیارا است؛ برای آدرس قطعی به مستندات لینک‌شده مراجعه کنید*

گزینه ۲: کرگدن (Kargadan)

کرگدن نیز NuGet را پوشش می‌دهد.

تنظیم:

```
dotnet nuget add source https://mirror.kargadan.ir/repository/nuget-group/index.json -n kargadan
```

Docker

برای دریافت ایمیج‌های Docker، چندین میرور داخلی وجود دارد. روش تنظیم در فایل /etc/docker/daemon.json (لینوکس) یا از طریق Docker Desktop (ویندوز) انجام می‌شود.

میرورهای پیشنهادی بر اساس اسکریپت iran-docker

```
docker.arvancloud.ir
docker.iranserver.com
docker.mobinhost.com
docker.kernel.ir
focker.ir
registry.docker.ir
docker.haiocloud.com
hub.mecan.ir
docker.nrp.co
docker-mirror.liara.ir
ghcr-mirror.liara.ir
mirrors.pardisco.co
mirror2.chabokan.net
docker.abrha.net
mirror-docker.runflare.com
```

روش تنظیم (لینوکس):

فایل /etc/docker/daemon.json را باز کنید (یا بسازید) و محتوای زیر را وارد کنید:

```
{
  "registry-mirrors": ["https://docker.arvancloud.ir"]
}
```

سپس سرویس Docker را ری‌استارت کنید:

```
sudo systemctl restart docker
```

روش تنظیم خودکار :می‌توانید از اسکریپت iran-dockerاستفاده کنید که به طور خودکار سریع‌ترین میرور را انتخاب و تنظیم می‌کند .

راهنمای استفاده از میرور Kargadanبرای زبان‌های برنامه‌نویسی و ابزارهای توسعه

میرور Kargadanعلاوه بر مخازن توزیع‌های لینوکس، امکان دریافت بسته‌ها و وابستگی‌های نرم‌افزاری را برای زبان‌های برنامه‌نویسی و ابزارهای توسعه فراهم می‌کند.با استفاده از این سرویس‌ها می‌توان وابستگی‌های پروژه‌ها را از یک آدرس جایگزین دریافت کرد و در برخی شرایط، سرعت نصب و به‌روزرسانی بسته‌ها را افزایش داد.

۱. – Python استفاده از مخزن PyPI

مدیر بسته Python یعنی pip از مخزن PyPIبرای دریافت کتابخانه‌ها استفاده می‌کند.

برای نصب یک کتابخانه از میرور :Kargadan

```
pip install --index-url \
https://mirror.kargadan.ir/repository/pypi-group/simple/ \
requests
```

در این مثال، کتابخانه requestsنصب می‌شود.

برای نصب دائمی و تنظیم آدرس میرور:

```
pip config set global.index-url \
https://mirror.kargadan.ir/repository/pypi-group/simple/
```

برای مشاهده تنظیمات:

```
pip config list
```

۲. – Conda استفاده از مخزن Conda-Forge

Condaیک مدیر بسته و محیط مجازی برای زبان‌هایی مانند Python است. مخزن Conda-Forgeنیز مجموعه بزرگی از بسته‌های متن‌باز را ارائه می‌کند.

برای نصب بسته NumPy:

```
conda install -c \
https://mirror.kargadan.ir/repository/conda-forge-proxy/ \
numpy
```

در این مثال، بسته numpy از کانال Conda-Forge نصب می‌شود.

۳. – Node.js استفاده از مخزن npm

npm مدیر بسته پیش‌فرض Node.js است و برای نصب کتابخانه‌های JavaScript استفاده می‌شود.

برای تغییر آدرس Registry:

```
npm config set registry \
https://mirror.kargadan.ir/repository/npm-group/
```

برای بررسی آدرس تنظیم‌شده:

```
npm config get registry
```

اکنون می‌توانید بسته‌ها را نصب کنید:

```
npm install express
```

برای بازگرداندن آدرس رسمی npm:

```
npm config set registry https://registry.npmjs.org/
```

۴. – Yarn استفاده از مخزن Yarn

Yarn یکی دیگر از مدیرهای بسته JavaScript است که برای مدیریت وابستگی‌های پروژه‌های Node.js استفاده می‌شود.

برای تنظیم Registry:

```
yarn config set registry \
https://mirror.kargadan.ir/repository/yarn-group/
```

برای مشاهده تنظیمات:

```
yarn config get registry
```

۵. – Docker تنظیم Registry Mirror

Dockerبرای دریافت Imageها از Docker Registryاستفاده می‌کند. با تنظیم Registry Mirrorمی‌توان درخواست‌های دریافت Imageرا از طریق میرور جایگزین انجام داد.

فایل تنظیمات Dockerرا باز کنید:

```
sudo nano /etc/docker/daemon.json
```

محتوای زیر را قرار دهید:

```
{
  "registry-mirrors": [
    "https://docker-mirror.kargadan.ir"
  ]
}
```

سپس سرویس Dockerرا راه‌اندازی مجدد کنید:

```
sudo systemctl restart docker
```

برای بررسی تنظیمات:

```
docker info
```

اکنون می‌توانید Imageدریافت کنید:

```
docker pull nginx
```

📌توجه کنید که Registry Mirrorمعمولاً برای دریافت Imageهای Docker Hubاستفاده می‌شود و جایگزین عمومی تمام Registryها، مانند GitHub Container Registry یا Quay، نیست.

۶. – Go استفاده از Go Modules Proxy

زبان Goاز Go Modulesبرای مدیریت وابستگی‌های پروژه استفاده می‌کند.

برای تنظیم میرور:

```
go env -w GOPROXY=\
https://mirror.kargadan.ir/repository/go-group/,direct
```

برای بررسی تنظیمات:

```
go env GOPROXY
```

در این تنظیم، ابتدا میرور Kargadanبررسی می‌شود و در صورت نیاز، گزینه directامکان مراجعه مستقیم به منبع اصلی را فراهم می‌کند.

۷. – Maven استفاده از مخزن Maven Central

Mavenابزار مدیریت وابستگی و ساخت پروژه‌های Javaاست.

برای استفاده از میرور Kargadan، فایل زیر را باز کنید:

```
nano ~/.m2/settings.xml
```

بخش زیر را داخل ساختار اصلی XMLقرار دهید:

```
<settings>
  <mirrors>
    <mirror>
      <id>kargadan</id>
      <url>https://mirror.kargadan.ir/repository/maven-central-group/</url>
      <mirrorOf>central</mirrorOf>
    </mirror>
  </mirrors>
</settings>
```

این تنظیم باعث می‌شود درخواست‌های مربوط به مخزن Maven Centralاز طریق میرور Kargadanانجام شوند.

۸. – .NET استفاده از مخزن NuGet

NuGetمدیر بسته رسمی اکوسیستم .NETاست و برای دریافت کتابخانه‌ها و وابستگی‌های پروژه‌های C#و .NETاستفاده می‌شود.

برای افزودن میرور:

```
dotnet nuget add source \
https://mirror.kargadan.ir/repository/nuget-group/index.json \
-n kargadan
```

برای مشاهده منابع ثبت‌شده:

```
dotnet nuget list source
```

برای نصب یک بسته:

```
dotnet add package Newtonsoft.Json
```

۹. – PHP استفاده از مخزن Composer

Composerمدیر وابستگی PHPاست و برای نصب و مدیریت کتابخانه‌های پروژه‌های PHPاستفاده می‌شود.

برای تنظیم میرور :Packagist

```
composer config -g \
repos.packagist composer \
https://mirror.kargadan.ir/repository/composer-jamko/
```

برای بررسی تنظیمات:

```
composer config -g --list
```

تنظیم میرورهای DockerHub

لیارا میرور داکر را در آدرس زیر، ارائه می‌دهد:

```
https://docker-mirror.liara.ir
```

برای تنظیم میرور فوق کافیست تا گام‌های زیر را طی کنید:

۱. تهیه بکاپ از تنظیمات فعلی

قبل از تنظیم میرورهای لیارا، بهتر است با اجرای دستور زیر، یک بکاپ از تنظیمات فعلی سیستم عامل خود تهیه کنید:

```
cp /etc/docker/daemon.json /etc/docker/daemon.json.bak
```

۲. تنظیم میرور لیارا

برای تنظیم میرور لیارا، دستور زیر را اجرا کنید:

```
echo '{
  "registry-mirrors": [
    "https://docker-mirror.liara.ir"
  ]
}' > /etc/docker/daemon.json
```

۳. ری‌استارت برنامه و فایل‌های کانفیگ

اکنون کافیست تا دستورات زیر را اجرا کنید تا از میرور لیارا، استفاده شود:

```
systemctl daemon-reload
systemctl restart docker
```

با انجام کارهای فوق، میرورهای لیارا در سیستم عامل شما تنظیم خواهند شد و درخواست‌های آپدیت یا دانلود پکیج به این میرورها ارسال خواهند شد.

سرویس میرور رجیستری گیت هاب

میرور رجیستری گیت هاب لیارا سرویسی است شامل ۹۹ درصد ایمیج‌های رجیستری Github Container که داخل ایران در دسترس است. مخزن گیت هاب لیارا درخواست‌های Pull شما را به سرورهای خود ارسال میکند و سپس ایمیج درخواستی شما را با سرعت بسیار بیشتر، بدون محدودیت‌های سخت گیرانه ghcr و مشکلات شبکه، در اختیار شما قرار میدهد.

تنظیم میرورهای Github Container Registry

لیارا میرور ghcr را در آدرس زیر، ارائه می‌دهد:

```
https://ghcr-mirror.liara.ir
```

برای تنظیم میرور فوق کافیست تا گام‌های زیر را طی کنید:

۱. تهیه بکاپ از تنظیمات فعلی

قبل از تنظیم میرورهای لیارا، بهتر است با اجرای دستور زیر، یک بکاپ از تنظیمات فعلی سیستم عامل خود تهیه کنید:

```
cp /etc/docker/daemon.json /etc/docker/daemon.json.bak
```

۲. تنظیم میرور لیارا

برای تنظیم میرور لیارا، دستور زیر را اجرا کنید:

```
echo '{
  "registry-mirrors": [
    "https://ghcr-mirror.liara.ir"
  ]
}' > /etc/docker/daemon.json
```

۳. ری‌استارت برنامه و فایل‌های کانفیگ

اکنون کافیست تا دستورات زیر را اجرا کنید تا از میرور لیارا، استفاده شود:

```
systemctl daemon-reload
systemctl restart docker
```

با انجام کارهای فوق، میرورهای لیارا در سیستم عامل شما تنظیم خواهند شد و درخواست‌های آپدیت یا دانلود پکیج به این میرورها ارسال خواهند شد.

سرویس میرور رجیستری Quay

میرور رجیستری Quay لیارا سرویسی است که بخش عمده‌ای از ایمیج‌های موجود در رجیستری quay.io را از طریق یک پروکسی سریع و داخل ایران در اختیار شما قرار می‌دهد. مخزن Quay لیارا درخواست‌های Pull را در ابتدا به سرورهای داخلی خود ارسال کرده و سپس ایمیج درخواستی را با سرعت بسیار بالاتر، بدون مواجهه با محدودیت‌های خارجی، تحریم‌ها و مشکلات اتصال بین‌المللی به شما ارائه می‌کند.

تنظیم میرورهای Quay

لیارا میرور Quay را در آدرس زیر، ارائه می‌دهد:

```
https://quay-mirror.liara.ir
```

برای تنظیم میرور فوق کافیست تا گام‌های زیر را طی کنید:

۱. تهیه بکاپ از تنظیمات فعلی

قبل از تنظیم میرورهای لیارا، بهتر است با اجرای دستور زیر، یک بکاپ از تنظیمات فعلی سیستم عامل خود تهیه کنید:

```
cp /etc/docker/daemon.json /etc/docker/daemon.json.bak
```

۲. تنظیم میرور لیارا

برای تنظیم میرور لیارا، دستور زیر را اجرا کنید:

```
echo '{
  "registry-mirrors": [
    "https://quay-mirror.liara.ir"
  ]
}' > /etc/docker/daemon.json
```

۳. ری‌استارت برنامه و فایل‌های کانفیگ

اکنون کافیست تا دستورات زیر را اجرا کنید تا از میرور لیارا، استفاده شود:

```
systemctl daemon-reload
systemctl restart docker
```

با انجام کارهای فوق، میرورهای لیارا در سیستم عامل شما تنظیم خواهند شد و درخواست‌های آپدیت یا دانلود پکیج به این میرورها ارسال خواهند شد.

سرویس میرور رجیستری Kubernetes

میرور رجیستری کوبرنتیز لیارا سرویسی است شامل ۹۹ درصد ایمیج‌های رجیستری K8S که داخل ایران در دسترس است. مخزن کوبرنتیز لیارا درخواست‌های Pull شما را به سرورهای خود ارسال می‌کند و سپس ایمیج درخواستی شما را با سرعت بسیار بیشتر، بدون مشکلات متعدد K8S و سایر مشکلات شبکه، در اختیار شما قرار می‌دهد.

تنظیم میرورهای K8S

لیارا میرور K8S را در آدرس زیر، ارائه می‌دهد:

```
https://k8s-mirror.liara.ir
```

برای تنظیم میرور فوق کافیست تا گام‌های زیر را طی کنید:

۱. تهیه بکاپ از تنظیمات فعلی

قبل از تنظیم میرورهای لیارا، بهتر است با اجرای دستور زیر، یک بکاپ از تنظیمات فعلی سیستم عامل خود تهیه کنید:

```
cp /etc/docker/daemon.json /etc/docker/daemon.json.bak
```

۲. تنظیم میرور لیارا

برای تنظیم میرور لیارا، دستور زیر را اجرا کنید:

```
echo '{
  "registry-mirrors": [
    "https://k8s-mirror.liara.ir"
  ]
}' > /etc/docker/daemon.json
```

۳. ری‌استارت برنامه و فایل‌های کانفیگ

اکنون کافیست تا دستورات زیر را اجرا کنید تا از میرور لیارا، استفاده شود:

```
systemctl daemon-reload
systemctl restart docker
```

با انجام کارهای فوق، میرورهای لیارا در سیستم عامل شما تنظیم خواهند شد و درخواست‌های آپدیت یا دانلود پکیج به این میرورها ارسال خواهند شد.

سرویس میرور PyPI لیارا

میرور PyPI لیارا یک مخزن داخلی از تمام پکیج‌های موجود در PyPI است که برای افزایش سرعت، پایداری و امنیت نصب پکیج‌های پایتون طراحی شده است. این سرویس درخواست‌های pip را ابتدا به سرورهای داخلی لیارا ارسال می‌کند و سپس نسخه‌ی موردنیاز پکیج را با سرعت بسیار بالاتر، بدون محدودیت‌های اینترنت بین‌الملل، مشکلات اتصال خارجی یا تحریم‌ها در اختیار شما قرار می‌دهد.

تنظیم میرورهای پای پی آی

لیارا میرور PyPI را در آدرس زیر، ارائه می‌دهد:

```
https://package-mirror.liara.ir/repository/pypi/
```

برای تنظیم میرور فوق کافیست تا دستور زیر را اجرا کنید:

```
export PIP_INDEX_URL=https://package-mirror.liara.ir/repository/pypi/simple
# for windows:
# setx PIP_INDEX_URL "https://package-mirror.liara.ir/repository/pypi/simple"
```

با انجام کار فوق، میرور لیارا برای شما تنظیم می‌شود. همچنین می‌توانید در آدرس زیر، لیست پکیج‌ها را مشاهده کنید:

```
https://package-mirror.liara.ir/service/rest/repository/browse/pypi/
```

سرویس میرور npm

میرور npm لیارا یک مخزن نرم‌افزاری شامل پکیج‌های npm است که برای افزایش سرعت، پایداری و امنیت در نصب پکیج‌های Node.jsطراحی شده است. این سرویس یک واسط برای دانلود پکیج‌های npm است و با ذخیره نسخه‌های پرمصرف در حافظه خود، فرآیند توسعه پروژه‌ها را به‌طور قابل‌توجهی افزایش می‌دهد. میرور ان پی ام لیارا درخواست‌های دانلود پکیج را ابتدا به سرورهای داخلی خود ارسال می‌کند و سپس نسخه مورد نیاز را بدون نیاز به اتصال بین‌المللی، بدون مواجهه با تحریم و بدون محدودیت‌های CDN خارجی با سرعتی بسیار بالاتر در اختیار شما قرار می‌دهد.

تنظیم میرورهای ان پی ام

لیارا میرور NPM را در آدرس زیر، ارائه می‌دهد:

```
https://package-mirror.liara.ir/repository/npm/
```

برای تنظیم میرور فوق کافیست تا دستور زیر را اجرا کنید:

```
npm config set registry https://package-mirror.liara.ir/repository/npm/ --global
```

برای بررسی میرورها، می‌توانید از دستور زیر استفاده کنید:

```
npm config get registry
```

با انجام کار فوق، میرور لیارا برای شما تنظیم می‌شود. همچنین می‌توانید در آدرس زیر، لیست پکیج‌ها را مشاهده کنید:

```
https://package-mirror.liara.ir/service/rest/repository/browse/npm/
```

بخش هشتم

سرویس‌های آپلود و اشتراک‌گذاری فایل ایرانی

نام سرویس	نیاز به ثبت‌نام	محدودیت رایگان گزارش‌شده	مدت نگهداری	توضیحات
Share4	خیر	۴۵۰ مگابایت	۳ روز	آپلود و اشتراک‌گذاری سریع بدون ثبت‌نام
Linklick	خیر	۱ گیگابایت	۷ روز	مناسب اشتراک‌گذاری موقت فایل
20Script	خیر	۱۰۰ مگابایت	نامشخص	سرویس ساده آپلود فایل بدون عضویت
8pic.ir	خیر	نامشخص	نامشخص	عمدتاً برای تصاویر؛ ادعای فضای نامحدود نیازمند تأیید است
UUpload (یوآپلود)	برای برخی امکانات بله	۱۰۰ مگابایت	نامشخص	گزارش‌هایی از ارائه فضای ۲۰ گیگابایتی برای کاربران ثبت‌نام‌شده وجود دارد
PicoFile	بله	۹۰۰ مگابایت برای کاربران مهمان در برخی گزارش‌ها	نامشخص	سرویس قدیمی اشتراک‌گذاری فایل؛ ظرفیت حساب و فایل باید تفکیک شود
Uploadkon	بله، با شماره موبایل	۱۰ گیگابایت	نامشخص	ظرفیت بالا؛ دارای محدودیت‌هایی برای برخی فرمت‌های فایل
BiaUpload	برای برخی امکانات بله	۲۰۰ مگابایت	نامشخص	عمدتاً مناسب آپلود تصاویر و فایل‌های کوچک
Files (فایلز)	بله	۲ گیگابایت	نامشخص	فضای ذخیره‌سازی و اشتراک‌گذاری فایل
آپارات	بله	محدودیت وابسته به شرایط حساب	وابسته به سیاست سرویس	پلتفرم میزبانی و اشتراک‌گذاری ویدئو؛ جایگزین آپلودسنتر عمومی نیست
Dardit	خیر	۱ گیگابایت	۷ روز	اشتراک‌گذاری موقت فایل بدون ثبت‌نام

بخش نهم

مدیریت Proxy و دسترسی شبکه در لینوکس

۱. مقدمه

در محیط‌های لینوکسی، دسترسی به اینترنت و دریافت بسته‌های نرم‌افزاری همیشه به‌صورت مستقیم انجام نمی‌شود. گاهی به دلیل محدودیت‌های شبکه، سیاست‌های سازمانی، مشکلات مسیریابی، اختلال در DNS یا نیاز به کنترل ترافیک، لازم است ارتباط برنامه‌ها با اینترنت از طریق یک سرور واسط برقرار شود.

یکی از راهکارهای متداول برای مدیریت این ارتباط، استفاده از Proxy است. پراکسی می‌تواند درخواست‌های کاربران و برنامه‌ها را دریافت کند و پس از ارسال آن‌ها به مقصد، پاسخ را به درخواست‌کننده بازگرداند.

در این فصل با انواع Proxy ، روش تنظیم آن در لینوکس، پیکربندی مدیرهای بسته و ابزارهای توسعه، استفاده از SOCKS5و روش‌های عیب‌یابی مشکلات شبکه آشنا می‌شویم.

۲. آشنایی با مفاهیم Proxy ، Mirror، Cache وRepository

این چهار مفهوم گاهی به‌جای یکدیگر استفاده می‌شوند، اما از نظر عملکرد تفاوت دارند.

۲,۱.Proxy

Proxyیک سرور واسط بین سیستم کاربر و مقصد شبکه است. هنگامی که یک برنامه درخواست اتصال به یک سرور را ارسال می‌کند، به‌جای برقراری ارتباط مستقیم، درخواست را به Proxy می‌فرستد.

Proxyدرخواست را دریافت می‌کند، ارتباط لازم را با مقصد برقرار می‌کند و پاسخ را به برنامه بازمی‌گرداند.

کاربردهای متداول Proxy عبارت‌اند از:

- عبور دادن درخواست‌ها از یک مسیر شبکه مشخص
- کنترل دسترسی کاربران به اینترنت
- ثبت و نظارت بر ترافیک شبکه
- اعمال سیاست‌های امنیتی
- استفاده از یک نقطه خروجی مشترک برای چند سیستم
- دسترسی به منابعی که از مسیر مستقیم در دسترس نیستند

۲,۲.Mirror

Mirrorنسخه‌ای از محتوا یا ساختار یک مخزن اصلی است که روی سرور دیگری نگهداری می‌شود.

برای مثال، یک Mirror از مخزن Debian می‌تواند بسته‌ها و اطلاعات مربوط به آن را در یک سرور جایگزین ارائه کند.

کاربردهای Mirror:

- کاهش فاصله شبکه‌ای تا مخزن
- افزایش سرعت دریافت بسته‌ها
- کاهش بار روی سرور اصلی
- فراهم کردن دسترسی جایگزین به مخازن نرم‌افزاری

Mirror الزاماً درخواست را به سرور اصلی ارسال نمی‌کند؛ معمولاً محتوا را از قبل دریافت و نگهداری می‌کند و به‌صورت مستقل در اختیار کاربران قرار می‌دهد.

۲,۳. Cache

Cache نسخه‌ای ذخیره‌شده از محتوای دریافت‌شده قبلی است. هنگامی که یک درخواست تکراری ارسال می‌شود، در صورت معتبر بودن محتوای ذخیره‌شده، Cache می‌تواند پاسخ را بدون دریافت مجدد آن از سرور اصلی ارائه کند.

برای مثال، اگر چند سیستم در یک شبکه سازمانی یک بسته یکسان را دانلود کنند، یک Cache Proxy می‌تواند از دریافت مکرر آن بسته از اینترنت جلوگیری کند.

۲,۴. Repository

Repository یا مخزن نرم‌افزاری، محلی برای نگهداری بسته‌های نرم‌افزاری و اطلاعات موردنیاز مدیر بسته‌ها است.

یک Repository معمولاً شامل موارد زیر است:

- فایل‌های بسته نرم‌افزاری
- اطلاعات نسخه‌ها
- وابستگی‌های نرم‌افزارها
- Metadata
- اطلاعات امضای دیجیتال و اعتبارسنجی

برای نمونه، APT در Debian و Ubuntu ، DNF در Fedora و RPM-based distributions و Pacman در Arch Linux از Repository برای دریافت بسته‌ها استفاده می‌کنند.

۲,۵. مقایسه مفاهیم

نمونه	عملکرد اصلی	مفهوم
Squid	واسطه‌گری ارتباط شبکه	**Proxy**
Debian Mirror	ارائه نسخه جایگزین از مخزن	**Mirror**
Squid Cache	ذخیره محتوای دریافت‌شده	**Cache**
APT Repository	نگهداری بسته‌ها وMetadata	**Repository**

این فناوری‌ها می‌توانند در کنار یکدیگر استفاده شوند. برای مثال، یک سازمان می‌تواند یک Repository داخلی داشته باشد که با یک Proxy به مخازن خارجی متصل می‌شود.

۳. انواع Proxy

پراکسی‌ها بر اساس پروتکل و نوع ارتباط به چند گروه تقسیم می‌شوند.

HTTP Proxy.۳,۱

HTTP Proxyبرای انتقال درخواست‌های HTTP استفاده می‌شود.

در این حالت، برنامه درخواست خود را به سرور Proxy ارسال می‌کند و Proxy آن را به مقصد انتقال می‌دهد.

نمونه آدرس:

```
http://proxy.example.com:8080
```

HTTPS Proxy.۳,۲

برای ارتباط HTTPS ، معمولاً از روش HTTP CONNECT استفاده می‌شود.

در این روش، برنامه از Proxy درخواست ایجاد یک تونل TCP به مقصد را می‌کند. پس از برقراری تونل، ارتباط TLSمستقیماً بین برنامه و سرور مقصد برقرار می‌شود.

نمونه آدرس:

```
http://proxy.example.com:8080
```

نکته: آدرس Proxy ممکن است با http://شروع شود، اما همچنان برای انتقال ارتباط HTTPS از آن استفاده شود. این موضوع با HTTPS Proxy واقعی که خود اتصال کلاینت به پراکسی را نیز با TLS محافظت می‌کند، یکسان نیست.

۳٫۳ SOCKS Proxy

SOCKSیک پروتکل واسط در سطح اتصال شبکه است و برخلافHTTP Proxy ، محدود به درخواست‌های HTTPنیست.

SOCKS5از قابلیت‌هایی مانند احراز هویت و انتقال ارتباطاتTCP پشتیبانی می‌کند. پشتیبانی از UDP نیز در این پروتکل تعریف شده است، اما همه کلاینت‌ها و سرویس‌های Proxy آن را پیاده‌سازی نمی‌کنند.

نمونه آدرس:

```
socks5://127.0.0.1:1080
```

در صورت استفاده ازSOCKS5 ، بهتر است در صورت پشتیبانی برنامه، روش Resolve کردن DNS از طریق Proxyنیز مشخص شود.

۴. تنظیم Proxy در محیط لینوکس

۲٫۲ .تعریف متغیرهای HTTP وHTTPS

برای تنظیم پروکسی در نشست فعلی ترمینال:

bash

export http_proxy="http://127.0.0.1:8080"

export https_proxy="http://127.0.0.1:8080"

export HTTP_PROXY="http://127.0.0.1:8080"

export HTTPS_PROXY="http://127.0.0.1:8080"

در این مثال، پروکسی روی سیستم محلی و پورت ۸۰۸۰ در حال اجرا فرض شده است. تعریف هم‌زمان حروف کوچک و بزرگ برای سازگاری با ابزارهای مختلف ضروری است، زیرا برخی برنامه‌ها فقط یکی از این حالت‌ها را تشخیص می‌دهند.

۲٫۳ .تعریف متغیر no_proxy

برای جلوگیری از عبور ترافیک داخلی از پروکسی، بهتر است no_proxyنیز تعریف شود:

```
export no_proxy="localhost,127.0.0.1,::1,10.0.0.0/8,192.168.0.0/16"
export NO_PROXY="$no_proxy"
```

این کار از بروز مشکلاتی مانند عدم دسترسی به سرویس‌های محلی یا کندی ارتباط با آدرس‌های داخلی جلوگیری می‌کند.

۲,۴ .ماندگار کردن تنظیمات

دستورات exportفقط روی نشست فعلی ترمینال اثر دارند. برای ماندگار کردن، خطوط زیر را در فایل پروفایل شل خود قرار دهید:

برای Bash فایل ~/.bashrcیا ~/.bash_profile :

```
export http_proxy="http://127.0.0.1:8080"
export https_proxy="http://127.0.0.1:8080"
export HTTP_PROXY="$http_proxy"
export HTTPS_PROXY="$https_proxy"
export no_proxy="localhost,127.0.0.1,::1"
export NO_PROXY="$no_proxy"
```

سپس دستور source ~/.bashrcرا اجرا کنید.

برای Zsh فایل :(~/.zshrcهمین خطوط را اضافه کرده و source ~/.zshrcرا اجرا کنید.

.تست عملکرد

پس از تنظیم، با دستورات زیر می‌توانید بررسی کنید که پروکسی اعمال شده است:

#بررسی متغیرها

```
env | grep -i proxy
```

#تست باcurl

```
curl -I https://example.com
```

#تست باwget

```
wget -qO- https://example.com | head
```

اگر پروکسی روی 127.0.0.1:8080در حال اجرا و درست تنظیم شده باشد، خروجی باید پاسخ سرور را نشان دهد؛ در غیر این صورت خطای اتصال دریافت می‌شود.

۴,۵. غیرفعال کردن Proxy

برای حذف تنظیمات از نشست فعلی:

```
unset http_proxy
unset https_proxy
unset HTTP_PROXY
unset HTTPS_PROXY
unset ALL_PROXY
unset NO_PROXY
```

۴,۶. دائمی کردن تنظیمات Proxy

اگر بخواهیم تنظیمات در نشست‌های بعدی نیز فعال باشند، می‌توانیم آن‌ها را در فایل تنظیمات Shell قرار دهیم.

برای Bash:

```
nano ~/.bashrc
```

در انتهای فایل اضافه می‌کنیم:

```
export http_proxy="http://127.0.0.1:8080"
export https_proxy="http://127.0.0.1:8080"
export HTTP_PROXY="http://127.0.0.1:8080"
export HTTPS_PROXY="http://127.0.0.1:8080"
export NO_PROXY="localhost,127.0.0.1,::1"
```

سپس تنظیمات را بارگذاری می‌کنیم:

```
source ~/.bashrc
```

این روش برای برنامه‌هایی که از محیط Shell اجرا می‌شوند مناسب است. سرویس‌های Systemd و برنامه‌هایی که توسط محیط گرافیکی یا سرویس‌های دیگر اجرا می‌شوند ممکن است این متغیرها را دریافت نکنند.

۵. تنظیم Proxy برای APT در Debian و Ubuntu

APT مدیر بسته توزیع‌های Debian ، Ubuntu و بسیاری از توزیع‌های مبتنی بر آن‌ها است.

برای تنظیم Proxy ، می‌توان یک فایل اختصاصی در مسیر تنظیمات APT ایجاد کرد.

```
sudo nano /etc/apt/apt.conf.d/80proxy
```

محتوای فایل:

```
Acquire::http::Proxy "http://127.0.0.1:8080";
Acquire::https::Proxy "http://127.0.0.1:8080";
```

پس از ذخیره فایل، می‌توان دریافت اطلاعات مخازن را آزمایش کرد:

```
sudo apt update
```

۵,۱. تنظیم Proxy با نام کاربری و رمز عبور

اگر Proxy نیازمند احراز هویت باشد، قالب آدرس به شکل زیر است:

```
http://USERNAME:PASSWORD@proxy.example.com:8080
```

نمونه:

```
Acquire::http::Proxy "http://user:password@127.0.0.1:8080";
Acquire::https::Proxy "http://user:password@127.0.0.1:8080";
```

ذخیره رمز عبور به‌صورت متن ساده در فایل تنظیمات می‌تواند خطر امنیتی ایجاد کند. در محیط‌های حساس، باید دسترسی فایل محدود شود و از روش‌های امن‌تر مدیریت اعتبارنامه استفاده شود.

۵,۲. غیرفعال کردن Proxy برای یک مخزن

در صورت نیاز می‌توان برای یک میزبان مشخص Proxy را غیرفعال کرد:

```
Acquire::http::Proxy::repository.example.com "DIRECT";
```

برای بازگرداندن APT به حالت مستقیم، فایل تنظیمات را حذف یا غیرفعال می‌کنیم:

```
sudo rm /etc/apt/apt.conf.d/80proxy
```

سپس:

```
sudo apt update
```

۶. تنظیم Proxy برای DNF در Fedora و توزیع‌های مبتنی بر RPM

DNFمدیر بسته در Fedora و بسیاری از توزیع‌های مبتنی بر RPM است.

فایل اصلی تنظیمات: DNF

```
/etc/dnf /dnf .conf
```

برای ویرایش:

```
sudo nano /etc/dnf /dnf .conf
```

در بخش [main]گزینه زیر را اضافه می‌کنیم:

```
[main]
proxy=http://127.0.0.1:8080
```

در صورت نیاز به احراز هویت:

```
[main]
proxy=http://proxy.example.com:8080
proxy_username=username
proxy_password=password
```

از ذخیره رمز عبور در فایل‌های قابل‌خواندن توسط کاربران غیرمجاز خودداری کنید.

برای آزمایش:

```
sudo dnf  clean all
sudo dnf  makecache
sudo dnf  repolist
```

۶,۱ تنظیم Proxy برای یک مخزن مشخص

اگر فقط بخواهیم یک Repository از Proxy استفاده کند، می‌توان گزینه مربوطه را در فایل همان مخزن قرار داد.

برای مثال:

```
[example-repository]
```

```
name=Example Repository
baseurl=https://repository.example.com/
enabled=1
gpgcheck=1
proxy=http://127.0.0.1:8080
```

۷. تنظیم Proxy برای Pacman در Arch Linux وManjaro

Pacmanمدیر بسته Arch Linux و توزیع‌هایی مانند Manjaro است.

فایل اصلی تنظیمات آن:

```
/etc/pacman.conf
```

Pacmanمانند APT و DNF گزینه عمومی مستقلی برای تعریف Proxy در فایل اصلی تنظیمات خود ندارد.
در حالت معمول، می‌توان از متغیرهای محیطی Proxy استفاده کرد.

برای نمونه:

```
export http_proxy="http://127.0.0.1:8080"
export https_proxy="http://127.0.0.1:8080"
```

سپس:

```
sudo pacman -Syyu
```

نکته مهم: متغیرهای محیطی ممکن است هنگام اجرای sudoمنتقل نشوند. برای اطمینان، می‌توان آن‌ها را به شکل کنترل‌شده به دستور منتقل کرد:

```
sudo env \
http_proxy="http://127.0.0.1:8080" \
https_proxy="http://127.0.0.1:8080" \
pacman -Syyu
```

روش دیگر استفاده از گزینه XferCommandدر pacman.confو ابزارهایی مانند curl یا wget است؛ با این حال، این روش باید با دقت تنظیم شود و ممکن است رفتار دانلود داخلی Pacman را تغییر دهد.

برای تنظیم Proxy درPacman ، نباید اعتبارسنجی امضای بسته‌ها را غیرفعال کرد.

۸. تنظیم Proxy برایGit

Gitبرای ارتباط با مخازن راه دور مانند GitHub و GitLab از پروتکل‌هایHTTP ، HTTPSو SSH استفاده می‌کند.

برای تنظیم Proxy در:Git

```
git config --global http.proxy \
http://127.0.0.1:8080
```

برای HTTPS نیز همین تنظیم استفاده می‌شود، زیرا Git تنظیم Proxy را در لایه HTTP مدیریت می‌کند.

برای مشاهده تنظیمات:

```
git config --global --get http.proxy
```

برای حذف تنظیمات:

```
git config --global --unset http.proxy
```

۸,۱. تنظیم Proxy فقط برای یک مخزن

می‌توان Proxy را فقط برای یک Repository تنظیم کرد:

```
git config http.proxy \
http://127.0.0.1:8080
```

این دستور باید داخل پوشه همان پروژه اجرا شود.

۸,۲. استفاده از SOCKS5 درGit

در صورت پشتیبانی نسخه Git و کتابخانه libcurl نصب‌شده از:SOCKS5

```
git config --global http.proxy \
socks5h://127.0.0.1:1080
```

این تنظیم برای عملیات Git از طریق HTTP و HTTPS است. ارتباط Git از طریق SSH به تنظیمات جداگانه SSH نیاز دارد.

۹. تنظیم Proxy برای Python و pip

مدیر بسته pip امکان تنظیم Proxy را از طریق پارامتر خط فرمان یا فایل تنظیمات فراهم می‌کند.

۹,۱. تنظیم Proxy هنگام نصب بسته

```
pip install requests \
--proxy http://127.0.0.1:8080
```

۹,۲. تنظیم دائمی Proxy

```
pip config set global.proxy \
http://127.0.0.1:8080
```

برای مشاهده:

```
pip config list
```

برای حذف:

```
pip config unset global.proxy
```

۹,۳. استفاده از Proxy همراه با Mirror

Proxy و Mirror دو مفهوم مستقل هستند. برای نمونه می‌توان درخواست‌های pip را از طریق Proxy ارسال کرد و در عین حال از یک Mirror مشخص برای دریافت بسته‌ها استفاده کرد:

```
pip install requests \
--index-url https://mirror.example.com/simple/ \
--proxy http://127.0.0.1:8080
```

در این مثال، Mirror منبع بسته‌ها است و Proxy مسیر ارتباطی را مشخص می‌کند.

۱۰. تنظیم Proxy برای npm

npm مدیر بسته Node.js است و تنظیمات Proxy را از طریق پیکربندی خود پشتیبانی می‌کند.

برای تنظیم Proxy:

```
npm config set proxy \
http://127.0.0.1:8080
```

برای HTTPS:

```
npm config set https-proxy \
http://127.0.0.1:8080
```

برای مشاهده:

```
npm config list
```

برای حذف تنظیمات:

```
npm config delete proxy
npm config delete https-proxy
```

۱۰,۱. ترکیب Proxy وRegistry

می‌توان Registry را روی یک Mirror تنظیم کرد و هم‌زمان از Proxy برای ارتباط شبکه استفاده کرد:

```
npm config set registry \
https://mirror.example.com/
npm config set https-proxy \
http://127.0.0.1:8080
```

Registryمشخص می‌کند بسته‌ها از کجا دریافت شوند و Proxy مسیر ارتباط را تعیین می‌کند.

۱۱. تنظیم Proxy برایDocker

Dockerدو نوع تنظیم Proxy مهم دارد:

۱. Proxy برایDocker Daemon

۲. Proxy برایContainer ها و عملیاتBuild

این دو تنظیم یکسان نیستند.

۱۱,۱. تنظیم Proxy برایDocker Daemon

Docker Daemonبرای دریافتImage ها و ارتباط باRegistry ها می‌تواند از Proxy استفاده کند.

در نسخه‌هایی که تنظیمات Proxy در daemon.jsonرا پشتیبانی می‌کنند، فایل زیر را ویرایش می‌کنیم:

```
sudo nano /etc/docker/daemon.json
```

نمونه:

```
{
 "proxies": {
  "http-proxy": "http://127.0.0.1:8080",
  "https-proxy": "http://127.0.0.1:8080",
  "no-proxy": "localhost,127.0.0.1,.local"
 }
}
```

پس از ذخیره فایل:

```
SUDO SYSTEMCTL RESTART DOCKER
```

این تنظیم را باید با نسخه Docker Engine و مستندات همان نسخه تطبیق داد.

۱۱,۲. تنظیم Proxy باSystemd

روش دیگر استفاده از فایل Drop-in برای سرویس Docker است:

```
sudo systemctl edit docker
```

محتوای زیر را وارد می‌کنیم:

```
[Service]
Environment="HTTP_PROXY=http://127.0.0.1:8080"
Environment="HTTPS_PROXY=http://127.0.0.1:8080"
Environment="NO_PROXY=localhost,127.0.0.1"
```

سپس:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

برای بررسی:

```
sudo systemctl show --property=Environment docker
```

۱۱,۳. تنظیم Proxy برایContainer ها

تنظیم Proxy برای Docker Daemon لزوماً به این معنا نیست که تمامContainer ها نیز همین تنظیم را دریافت می‌کنند.

برای تنظیم Proxy در زمان اجرای:Container

```
docker run --rm \
-e HTTP_PROXY=http://proxy.example.com:8080 \
-e HTTPS_PROXY=http://proxy.example.com:8080 \
nginx
```

برای Build نیز می‌توان از Build Arguments استفاده کرد، البته نحوه مصرف آن‌ها به Dockerfile و ابزار Buildبستگی دارد.

۱۲. استفاده از SOCKS5 در لینوکس

SOCKS5یکی از روش‌های متداول انتقال ارتباطات از طریق سرور واسط است.

فرض کنیم یک Proxy SOCKS5 روی آدرس زیر در حال اجرا است:

```
127.0.0.1:1080
```

۱۲,۱. تنظیم SOCKS5 برای نشست ترمینال

```
export ALL_PROXY="socks5h://127.0.0.1:1080"
```

برای ابزارهایی که از این متغیر پشتیبانی می‌کنند، ارتباط از طریق SOCKS5 برقرار خواهد شد.

۱۲,۲. آزمایش ارتباط باSOCKS5

```
curl --socks5-hostname 127.0.0.1:1080 \
https://example.com
```

گزینه --socks5-hostnameباعث می‌شود Resolve کردن نام دامنه نیز از طریق Proxy انجام شود.

۱۲,۳. تفاوت socks5 وsocks5h

نوع	محل Resolve شدنDNS
socks5	معمولاً در سمت کلاینت
socks5h	در سمتProxy

اگر هدف جلوگیری از ارسال درخواست DNS به Resolver محلی باشد، استفاده از socks5hدر برنامه‌های پشتیبانی‌کننده مناسب‌تر است.

۱۲,۴. محدودیت استفاده ازSOCKS5

همه برنامه‌ها از SOCKS5 پشتیبانی نمی‌کنند. همچنین برخی ابزارها فقط HTTP Proxy را می‌پذیرند.

برای برنامه‌هایی که SOCKS5 را مستقیماً پشتیبانی نمی‌کنند، می‌توان از یک واسط تبدیل SOCKS به HTTPمانند Privoxy استفاده کرد.

Privoxyمی‌تواند درخواست‌های HTTP را از یک مسیر SOCKS عبور دهد، اما باید جداگانه نصب و پیکربندی شود.

۱۳. بررسی محدودیت‌های DNS وTLS

تنظیم Proxy همیشه به معنی برطرف شدن تمام مشکلات اتصال نیست. بعضی خطاها بهDNS ، گواهی‌های TLSیا مسیر شبکه مربوط هستند.

۱۳,۱. بررسیDNS

برای بررسی Resolve شدن یک دامنه:

```
getent hosts example.com
```

یا:

```
nslookup example.com
```

در سیستم‌هایی که ابزار digنصب است:

```
dig example.com
```

اگر دامنه Resolve نشود، ممکن است مشکل از DNS محلی، تنظیمات شبکه یا دسترسی به Resolver باشد.

۱۳,۲. بررسی DNS از طریقSOCKS5

برای آزمایش ارتباطی که DNS نیز از Proxy عبور می‌کند:

```
curl --socks5-hostname 127.0.0.1:1080 \
https://example.com
```

اگر درخواست با این روش موفق باشد، اما Resolve محلی دامنه شکست بخورد، احتمال وجود مشکل در مسیر DNSمحلی مطرح می‌شود.

۱۳,۳. بررسیTLS

TLSبرای ایجاد ارتباط رمزگذاری‌شده و اعتبارسنجی هویت سرور استفاده می‌شود.

برای بررسی گواهی و ارتباط TLS:

```
openssl s_client \
-connect example.com:443 \
-servername example.com
```

برای بررسی جزئیات ارتباط HTTPS:

```
curl -Iv https://example.com
```

خطاهای متداول TLS عبارت‌اند از:

- گواهی منقضی‌شده
- نام دامنه نامعتبر در گواهی
- نبودن گواهی CA مورد اعتماد
- ساعت و تاریخ نادرست سیستم
- جایگزینی گواهی توسط Proxy سازمانی
- ناسازگاری نسخه TLS یا تنظیمات رمزنگاری

۱۴. عیب‌یابی مشکلات Proxy

در زمان استفاده از Proxy ممکن است خطاهای مختلفی رخ دهد.

خطا	علت احتمالی	راهکار
Connection refused	Proxy اجرا نمی‌شود یا پورت اشتباه است	بررسی سرویس و پورت
Connection timed out	مسیر شبکه یا فایروال ارتباط را مسدود کرده است	بررسی Firewall و Route
407 Proxy Authentication Required	اطلاعات احراز هویت اشتباه است یا ارسال نشده است	بررسی نام کاربری و رمز عبور
403 Forbidden	دسترسی به مقصد توسط Proxy محدود شده است	بررسی سیاست‌های دسترسی
Could not resolve host	مشکل DNS یا Resolve دامنه	بررسی Resolver
Certificate verification failed	مشکل گواهی TLS یا CA	بررسی گواهی و ساعت سیستم
Proxy CONNECT aborted	Proxy نتوانسته تونل HTTPS ایجاد کند	بررسی پشتیبانی CONNECT و دسترسی مقصد

بخش دهم

آموزش تنظیم NTPو سرورهای زمان در ایران در Linuxو Windows

۱. مقدمه

پروتکل NTP (Network Time Protocol)برای همگام‌سازی ساعت سیستم‌ها با سرورهای زمانی شبکه استفاده می‌شود. تنظیم صحیح زمان در سیستم‌عامل‌ها و سرورها اهمیت زیادی دارد؛ زیرا بسیاری از سرویس‌های شبکه و امنیتی به ساعت دقیق وابسته‌اند.

برای مثال، اختلاف ساعت سیستم می‌تواند باعث بروز مشکلات زیر شود:

- نامعتبر شناخته‌شدن گواهی‌های SSL/TLS
- ثبت نادرست زمان رویدادها در لاگ‌ها
- اختلال در احراز هویت Kerberosو سرویس‌های دامنه
- ناهماهنگی در مانیتورینگ و تحلیل رخدادهای امنیتی
- ثبت اشتباه زمان اجرای وظایف زمان‌بندی‌شده
- مشکل در همگام‌سازی پایگاه‌های داده و سرویس‌های توزیع‌شده

در شرایطی که دسترسی به سرورهای زمانی خارجی محدود یا ناپایدار است، استفاده از سرورهای NTPدر داخل ایران می‌تواند گزینه مناسبی باشد؛ البته باید دسترس‌پذیری و صحت عملکرد هر سرور پیش از استفاده بررسی شود.

۲. NTP چگونه کار می‌کند؟

NTPساعت سیستم را با استفاده از تبادل پیام‌های زمانی با یک یا چند سرور تنظیم می‌کند. کلاینت NTP اختلاف ساعت خود با سرور را محاسبه می‌کند و به‌تدریج یا در شرایط لازم ساعت سیستم را اصلاح می‌کند.

در معماری معمول:

۱. سیستم کاربر یا سرور، نقش NTP Clientرا دارد.
۲. سرور زمانی، مرجع ارائه زمان است.
۳. کلاینت از طریق UDP Port 123با سرور ارتباط برقرار می‌کند.
۴. زمان دریافت‌شده بررسی و با ساعت محلی مقایسه می‌شود.
۵. سیستم، اختلاف ساعت را اصلاح می‌کند.

نکته NTP :معمولاً از پورت UDPشماره 123استفاده می‌کند. بنابراین، بازبودن این پورت در مسیر شبکه و فایروال برای همگام‌سازی ضروری است.

۳. سرورهای زمانی داخل ایران

برای استفاده از NTPداخلی، بهتر است سروری انتخاب شود که از نظر شبکه‌ای به زیرساخت شما نزدیک باشد، به‌صورت پایدار پاسخ دهد و وضعیت همگام‌سازی آن قابل بررسی باشد.

در انتخاب سرور به این موارد توجه کنید:

- پاسخ‌گویی روی UDP/123
- پایداری و دسترس‌پذیری
- دقت و وضعیت همگام‌سازی سرور
- امکان استفاده از چند سرور مستقل
- سیاست استفاده و محدودیت نرخ درخواست‌ها

توجه :نام دامنه یا قرارداشتن یک سرور در ایران به‌تنهایی تضمین نمی‌کند که سرویس NTPعمومی و فعال ارائه می‌دهد. پیش از درج آدرس در تنظیمات سیستم، آن را آزمایش کنید.

فهرست سرورهای پیشنهادی برای آزمایش

سرور	توضیحات
ntp.ultratime.ir	سرویس NTPایرانی که ارائه‌دهنده، استفاده عمومی از UDP/123را اعلام کرده است. الترا تایم
ntp.day.ir	به‌عنوان سرور زمانی ایرانی معرفی شده است؛ پیش از استفاده عملی تست شود. سرور دات آی آر
ntp.cloud.ir	به‌عنوان سرور زمانی ایرانی معرفی شده است؛ نیازمند بررسی دسترسی است. سرور دات آی آر
ntp.time.ir	به‌عنوان سرور زمانی ایرانی معرفی شده است؛ نیازمند بررسی دسترسی است. سرور دات آی آر

برای استفاده از چند منبع، می‌توانید از این نشانی‌های NTP Poolنیز استفاده کنید:

```
0.asia.pool.ntp.org
1.asia.pool.ntp.org
2.asia.pool.ntp.org
3.asia.pool.ntp.org
```

نکته مهم درباره NTP Poolایران :در زمان بررسی، پروژه NTP Poolاعلام کرده است که تعداد سرورهای فعال در منطقه ایران کافی نیست و استفاده از منطقه آسیا را پیشنهاد می‌کند. بنابراین بهتر است ir.pool.ntp.orgرا تنها منبع زمانی زیرساخت خود قرار ندهید.

```
ntppool.org
```

منابع بالا از نظر معرفی سرویس بررسی شده‌اند؛ این موضوع به معنی تأیید مستقل کیفیت، دقت یا دسترس‌پذیری دائمی آن‌ها نیست.

۴. آزمایش دسترسی به سرور NTP

قبل از تغییر تنظیمات سیستم، بهتر است بررسی کنید سرور انتخاب‌شده پاسخ می‌دهد یا خیر.

در لینوکس، اگر Chronyنصب باشد، دستور زیر برای آزمایش مناسب است:

```
chronyd -Q -t 5 'Server ntp.ultratime.ir iburst maxsamples 1'
```

این دستور برای بررسی زمان دریافتی است و تنظیم دائمی سرویس را انجام نمی‌دهد.

در ویندوز، CMDیا PowerShellرا باز کنید:

```
w32tm /stripchart /computer:ntp.ultratime.ir /dataonly /samples:10
```

در صورت دسترسی موفق، نمونه‌هایی از اختلاف زمانی با سرور نمایش داده می‌شود.

برای بررسی DNSنیز می‌توانید از دستور زیر استفاده کنید:

```
nslookup ntp.ultratime.ir
```

توجه داشته باشید که موفقیت pingبه‌تنهایی نشان‌دهنده فعال‌بودن NTPنیست؛ زیرا NTPاز UDP/123 استفاده می‌کند و ممکن است ICMPمسدود باشد.

۵. تنظیم NTPدر Debianو Ubuntuبا Chrony

در توزیع‌های جدید لینوکس، Chronyیکی از گزینه‌های مناسب برای مدیریت همگام‌سازی زمان است.

ابتدا آن را نصب کنید:

```
sudo apt update
sudo apt install chrony -y
```

فایل تنظیمات را باز کنید:

```
sudo nano /etc/chrony/chrony.conf
```

سرورهای زمانی قبلی را بررسی کنید. در صورت نیاز، آن‌ها را غیرفعال کرده و منابع زمانی موردنظر را اضافه کنید:

```
Server ntp.ultratime.ir iburst
Server 0.asia.pool.ntp.org iburst
Server 1.asia.pool.ntp.org iburst
```

پارامتر iburstباعث می‌شود هنگام شروع ارتباط، درخواست‌های اولیه با فاصله زمانی کوتاه‌تری ارسال شوند تا همگام‌سازی سریع‌تر انجام شود.

فایل را ذخیره کنید و سرویس را راه‌اندازی مجدد کنید:

```
sudo systemctl enable chrony
sudo systemctl restart chrony
```

وضعیت سرویس را بررسی کنید:

```
sudo systemctl status chrony
```

برای مشاهده منابع زمانی:

```
chronyc sources -v
```

برای مشاهده وضعیت کلی همگام‌سازی:

```
chronyc tracking
```

در خروجی chronyc sources -vعلامت‌های مهم عبارت‌اند از:

علامت	مفهوم
^*	منبع زمانی انتخاب‌شده و مورد استفاده
^+	منبعی که در همگام‌سازی مشارکت دارد
^-	منبع معتبر، اما فعلاً انتخاب‌نشده
^?	منبعی که هنوز قابل دسترسی یا معتبر تشخیص داده نشده است
^x	منبعی که به‌عنوان منبع نامعتبر تشخیص داده شده است

در حالت مطلوب، حداقل یک منبع با علامت ^*مشاهده می‌شود.

۶. تنظیم NTP در Debian و Ubuntu با systemd-timesyncd

اگر سیستم شما از Chrony استفاده نمی‌کند و سرویس systemd-timesyncd فعال است، می‌توانید تنظیمات را از طریق این سرویس انجام دهید.

ابتدا وضعیت سرویس را بررسی کنید:

```
timedatectl status
```

سپس فایل تنظیمات را باز کنید:

```
sudo nano /etc/systemd/timesyncd.conf
```

مقادیر زیر را تنظیم کنید:

```
[Time]
NTP=ntp.ultratime.ir 0.asia.pool.ntp.org 1.asia.pool.ntp.org
FallbackNTP=2.asia.pool.ntp.org 3.asia.pool.ntp.org
```

سرویس را فعال و راه‌اندازی مجدد کنید:

```
sudo systemctl enable systemd-timesyncd
sudo systemctl restart systemd-timesyncd
```

برای فعال‌کردن همگام‌سازی خودکار:

```
sudo timedatectl set-ntp true
```

وضعیت همگام‌سازی را مشاهده کنید:

```
timedatectl timesync-status
```

همچنین می‌توانید از دستور زیر استفاده کنید:

```
timedatectl status
```

نکته :برای جلوگیری از تداخل، فقط یک سرویس اصلی همگام‌سازی زمان مانند Chrony یا systemd-timesyncd را به‌عنوان مدیر ساعت سیستم فعال نگه دارید.

۷. تنظیم NTP در Fedora، Rocky Linux و AlmaLinux

در توزیع‌های مبتنی بر Red Hat معمولاً از Chrony استفاده می‌شود.

نصب بسته:

```
sudo dnf  install chrony -y
```

ویرایش فایل تنظیمات:

```
sudo nano /etc/chrony.conf
```

منابع زمانی را اضافه کنید:

```
Server ntp.ultratime.ir iburst
Server 0.asia.pool.ntp.org iburst
Server 1.asia.pool.ntp.org iburst
```

فعال‌سازی سرویس:

```
sudo systemctl enable --now chronyd
```

راه‌اندازی مجدد پس از تغییر تنظیمات:

```
sudo systemctl restart chronyd
```

بررسی وضعیت:

```
chronyc tracking
chronyc sources -v
```

در صورت استفاده از فایروال، ترافیک خروجی UDP/123 باید مجاز باشد.

۸. تنظیم NTP در Arch Linux و Manjaro

در Arch Linux و Manjaro می‌توانید از Chrony یا systemd-timesyncd استفاده کنید.

برای نصب Chrony:

```
sudo pacman -S chrony
```

فایل تنظیمات را باز کنید:

```
sudo nano /etc/chrony.conf
```

این منابع را اضافه کنید:

```
Server ntp.ultratime.ir iburst
Server 0.asia.pool.ntp.org iburst
Server 1.asia.pool.ntp.org iburst
```

سپس سرویس را فعال کنید:

```
sudo systemctl enable --now chronyd
```

برای بررسی وضعیت:

```
chronyc tracking
chronyc sources -v
```

اگر قصد استفاده از systemd-timesyncdرا دارید، ابتدا مطمئن شوید سرویس دیگری مانند Chronyهم‌زمان مدیریت ساعت را بر عهده ندارد.

۹. تنظیم ساعت و NTPدر Windows 10وWindows 11

در ویندوز می‌توان سرور زمانی را از طریق تنظیمات گرافیکی یا خط فرمان تغییر داد.

ابتدا Command Promptیا PowerShellرا با دسترسی Administratorباز کنید.

تنظیم سرور NTP:

```
w32tm /config /manualpeerlist:"ntp.ultratime.ir,0x8" /syncfromflags:manual /update
```

در این دستور:

- manualpeerlistسرور زمانی موردنظر را مشخص می‌کند.
- 0x8باعث استفاده از حالت Clientدر Windows Timeمی‌شود.
- syncfromflags:manualمنابع زمانی دستی را انتخاب می‌کند.

سرویس Windows Timeرا راه‌اندازی مجدد کنید:

```
net stop w32time
```

```
net start w32time
```

سپس همگام‌سازی را اجرا کنید:

```
w32tm /resync
```

برای مشاهده سرور زمانی فعال:

```
w32tm /query /source
```

برای مشاهده وضعیت کامل:

```
w32tm /query /status
```

برای مشاهده تنظیمات فعلی:

```
w32tm /query /configuration
```

در صورت نیاز به استفاده از چند سرور زمانی:

```
w32tm /config /manualpeerlist:"ntp.ultratime.ir,0x8 0.asia.pool.ntp.org,0x8 1.asia.pool.ntp.org,0x8" /syncfromflags:manual /update
```

۱۰. تنظیم NTP در Windows Server

در شبکه‌های سازمانی، بهتر است یک Windows Server به‌عنوان منبع زمانی داخلی تعریف شود و کلاینت‌ها و سرورهای عضو دامنه از آن زمان دریافت کنند.

برای تنظیم یک سرور به‌عنوان منبع زمانی معتبر، PowerShell را با دسترسی Administrator اجرا کنید:

```
w32tm /config /manualpeerlist:"ntp.ultratime.ir,0x8" /syncfromflags:manual /reliable:YES /update
```

سپس:

```
net stop w32time
net start w32time
w32tm /resync
```

برای مشاهده وضعیت:

```
w32tm /query /status
```

در محیط Active Directory، طراحی سلسله‌مراتبی زمان اهمیت دارد. معمولاً PDC Emulatorدر ریشه جنگل دامنه به منبع زمانی خارجی یا معتبر متصل می‌شود و سایر Domain Controllerها و کلاینت‌ها از سلسله‌مراتب دامنه زمان دریافت می‌کنند.

هشدار :گزینه /reliable:YESرا برای هر کلاینت یا عضو عادی دامنه استفاده نکنید. این گزینه برای سروری است که عمداً به‌عنوان منبع زمانی معتبر سازمان تعریف شده است.

۱۱. تنظیم منطقه زمانی ایران (Asia/Tehran)

یکی از اشتباهات رایج این است که تصور می‌شود NTPساعت ایران را مستقیماً ارسال می‌کند.

در واقع، NTPزمان را بر اساس UTCمنتقل می‌کند و سیستم‌عامل با استفاده از منطقه زمانی، ساعت محلی را نمایش می‌دهد.

برای تنظیم منطقه زمانی ایران در لینوکس:

```
sudo timedatectl set-timezone Asia/Tehran
```

بررسی منطقه زمانی:

```
timedatectl
```

نمایش ساعت:

```
date
```

در ویندوز، برای تنظیم منطقه زمانی ایران می‌توانید از دستور زیر استفاده کنید:

```
tzutil /s "Iran Standard Time"
```

برای مشاهده منطقه زمانی فعلی:

```
tzutil /g
```

برای مشاهده فهرست مناطق زمانی:

```
tzutil /l
```

نکته مهم این است که ایران از سال ۱۴۰۱ تغییر ساعت فصلی را کنار گذاشته است. با این حال، برای اعمال صحیح قواعد زمانی باید بسته اطلاعات منطقه زمانی سیستم‌عامل، مانند tzdataدر لینوکس، به‌روز باشد.

۱۲. تنظیم NTP در MikroTik

در روترهای MikroTik نیز می‌توان سرور زمانی را تنظیم کرد تا ساعت دستگاه و زمان ثبت لاگ‌ها دقیق باشد.

در RouterOS نسخه‌های جدید، از دستورات زیر استفاده کنید:

```
/system ntp client set enabled=yes
```

افزودن سرور زمانی:

```
/system ntp client servers add address=ntp.ultratime.ir
```

افزودن یک منبع پشتیبان:

```
/system ntp client servers add address=0.asia.pool.ntp.org
```

مشاهده وضعیت:

```
/system ntp client print
```

برای تنظیم منطقه زمانی:

```
/system clock set time-zone-name=Asia/Tehran
```

در صورت استفاده از نسخه‌های قدیمی RouterOS، ساختار دستورات NTP ممکن است متفاوت باشد.

۱۳. ساخت سرور NTP داخلی برای شبکه سازمانی

در شبکه‌های بزرگ، بهتر است تمام کلاینت‌ها مستقیماً به اینترنت متصل نشوند. می‌توان یک سرور لینوکسی را به‌عنوان NTP Server داخلی تنظیم کرد.

در این معماری:

- سرور مرکزی از NTPهای معتبر دریافت زمان می‌کند.
- سرور مرکزی زمان را در شبکه داخلی منتشر می‌کند.
- کلاینت‌های لینوکس و ویندوز از سرور داخلی زمان می‌گیرند.
- در صورت اختلال اینترنت، کلاینت‌ها همچنان می‌توانند از آخرین وضعیت زمانی سرور داخلی استفاده کنند؛ اما دقت زمان به پایداری ساعت سرور و مدت قطع ارتباط وابسته خواهد بود.

برای پیاده‌سازی با Chrony، روی سرور مرکزی فایل زیر را ویرایش کنید:

```
sudo nano /etc/chrony/chrony.conf
```

منابع زمانی بالادستی را تعریف کنید:

```
Server ntp.ultratime.ir iburst
Server 0.asia.pool.ntp.org iburst
Server 1.asia.pool.ntp.org iburst
```

سپس محدوده شبکه داخلی را مجاز کنید. برای مثال، اگر شبکه سازمانی شما 192.168.10.0/24است:

```
allow 192.168.10.0/24
```

سرویس را راه‌اندازی مجدد کنید:

```
sudo systemctl restart chrony
```

در فایروال سرور مرکزی، دسترسی ورودی UDP/123را فقط برای شبکه داخلی مجاز کنید.

برای نمونه، در :UFW

```
sudo ufw allow from 192.168.10.0/24 to any port 123 proto udp
```

اکنون کلاینت‌های شبکه می‌توانند به‌جای سرورهای عمومی، از IPسرور داخلی استفاده کنند.

برای مثال، در فایل تنظیمات Chronyکلاینت‌ها:

```
Server 192.168.10.10 iburst
```

در این مثال، 192.168.10.10آدرس IPسرور NTPداخلی است.

نکته امنیتی :سرور NTPداخلی را بدون نیاز به همه شبکه‌ها یا اینترنت عمومی منتشر نکنید. دسترسی سرویس را به محدوده‌های مورد اعتماد محدود کنید.

۱۴. عیب‌یابی مشکلات NTP

مشکل	علت احتمالی	راه‌حل
Timeout	مسدودبودن UDP/123	بررسی فایروال و مسیر شبکه
عدم Resolveدامنه	مشکل DNS	بررسی nslookupیا dig
ساعت همچنان اشتباه است	منطقه زمانی نادرست	تنظیم Asia/Tehran
سرویس فعال نیست	سرویس NTPاجرا نشده	بررسی systemd
منابع زمانی نامعتبرند	سرور پاسخ نامناسب می‌دهد	انتخاب منابع دیگر
اختلاف ساعت زیاد است	ساعت محلی بسیار عقب یا جلو است	بررسی وضعیت و اصلاح اولیه ساعت
Windows Resync Error	مشکل سرویس Windows Timeیا ارتباط شبکه	بررسی w32tmو سرویس

دستورهای مفید برای عیب‌یابی در لینوکس:

```
timedatectl status
systemctl status chrony
chronyc tracking
chronyc sources -v
journalctl -u chrony --since today
```

در توزیع‌هایی که نام سرویس chronydاست:

```
journalctl -u chronyd --since today
```

۱۵. نکات امنیتی و عملیاتی

- از چند منبع زمانی مستقل استفاده کنید تا خرابی یک سرور باعث توقف همگام‌سازی نشود.
- از سرورهای ناشناخته و بدون سابقه مشخص به‌عنوان منبع اصلی زمان استفاده نکنید.
- در شبکه‌های سازمانی، NTPرا از طریق یک سرور داخلی مدیریت کنید.
- برای کلاینت‌های عادی فقط دسترسی لازم به UDP/123را فراهم کنید.
- ساعت سیستم را با منطقه زمانی اشتباه نگیرید؛ NTPو Time Zoneدو تنظیم مستقل هستند.
- وضعیت همگام‌سازی را به‌صورت دوره‌ای بررسی کنید.
- از اجرای هم‌زمان چند سرویس مدیریت زمان خودداری کنید.
- در زیرساخت‌های حساس، استفاده از NTS (Network Time Security)را در صورت پشتیبانی کلاینت و سرور و امکان دسترسی شبکه بررسی کنید.

۱۶. جمع‌بندی دستورات مهم

عملیات	دستور
مشاهده وضعیت زمان لینوکس	timedatectl status
تنظیم منطقه زمانی ایران	timedatectl set-timezone Asia/Tehran
فعال‌کردن NTPدر systemd	timedatectl set-ntp true
مشاهده منابع Chrony	chronyc sources -v
مشاهده وضعیت Chrony	chronyc tracking
تنظیم منطقه زمانی ویندوز	tzutil /s "Iran Standard Time"
مشاهده سرور NTPویندوز	w32tm /query /source
همگام‌سازی دستی ویندوز	w32tm /resync
تست سرور NTPویندوز	w32tm /stripchart /computer:ntp.ultratime.ir /dataonly /samples:10

نتیجه :برای سرورهای مستقر در ایران، استفاده از یک NTPداخلی معتبر به همراه یک یا چند منبع پشتیبان می‌تواند وابستگی به مسیرهای بین‌المللی را کاهش دهد. در شبکه‌های سازمانی نیز راه‌اندازی یک NTP Server مرکزی، مدیریت زمان و عیب‌یابی را ساده‌تر می‌کند.

بخش یازدهم

آسان‌دسک؛ نرم‌افزار دسترسی و پشتیبانی از راه دور

آسان‌دسک (AsanDesk)نرم‌افزاری برای برقراری ارتباط از راه دور و ارائه خدمات پشتیبانی است که نسخه‌هایی برای سیستم‌عامل‌های مختلف ارائه می‌دهد. چنین ابزارهایی می‌توانند برای پشتیبانی فنی، مدیریت سیستم‌ها و کمک به کاربران بدون نیاز به حضور فیزیکی در محل مورد استفاده قرار گیرند.

بر اساس اطلاعات صفحه دانلود آسان‌دسک، نسخه 1.5.0برای سیستم‌عامل‌های زیر معرفی شده است:

- Windows نسخه‌های ۶۴ بیتی و ۳۲ بیتی
- Linux :نسخه‌های سازگار با Ubuntuو Debian
- Android :نسخه مخصوص دستگاه‌های اندرویدی
- macOS :نسخه مخصوص سیستم‌عامل مک
- سایر نسخه‌ها :از طریق بخش دانلود وب‌سایت

آموزش تنظیم آسان‌دسک برای اتصال به سرور ایران

آسان‌دسک (AsanDesk) امکان اتصال از راه دور و پشتیبانی فنی را فراهم می‌کند. برای استفاده از سرور ایران، لازم است تنظیمات سرور اختصاصی نرم‌افزار را مطابق راهنمای رسمی آن بررسی و در صورت نیاز اصلاح کنید.

مراحل تنظیم آسان‌دسک

مرحله اول: بررسی وضعیت نرم‌افزار

نرم‌افزار آسان‌دسک را اجرا کنید و مطمئن شوید وضعیت نمایش‌داده‌شده در پایین پنجره روی Readyقرار دارد. سپس از قسمت بالای نرم‌افزار، در سمت راست، وارد بخش تنظیمات شوید.

مرحله دوم: بررسی تنظیمات سرور اختصاصی

در بخش تنظیمات، بررسی کنید اطلاعات سرور اختصاصی آسان‌دسک به‌درستی وارد شده باشند. طبق راهنمای رسمی، این اطلاعات باید در نسخه‌های ویندوز ۳۲ بیتی و ۶۴ بیتی به‌صورت پیش‌فرض تنظیم شده باشند.

اگر مقادیر Keyو Hostدر قسمت مربوط وجود ندارند، مراحل بعدی را انجام دهید.

مرحله سوم: ورود به تنظیمات شبکه

۱. تب Networkرا انتخاب کنید.

۲. اگر دکمه آبی‌رنگ Unlock network settingsنمایش داده شد، روی آن کلیک کنید.

۳. گزینه Server ID/Relay را انتخاب کنید.

مرحله چهارم: واردکردن اطلاعات سرور ایران

مقادیر زیر را در فیلدهای مربوط وارد کنید:

- **ID Server (Host):**

```
rd.asandesk.ir
```

- **Key:**

```
joRoqzc0SDOHe61V9WrRORk4PqBVzBntfGtzveM3h8M=
```

پس از واردکردن اطلاعات، تنظیمات را ذخیره کنید و در صورت نیاز نرم‌افزار را مجدداً اجرا کنید.

نکات مهم

- اگر از نسخه ویندوز استفاده می‌کنید، ابتدا تنظیمات پیش‌فرض را بررسی کنید؛ ممکن است نیازی به تغییر آن‌ها نباشد.
- اطلاعات Hostو Keyرا دقیقاً مطابق راهنمای رسمی وارد کنید.
- در صورت برقرارنشدن اتصال، وضعیت شبکه، دسترسی به سرور و تنظیمات نرم‌افزار را بررسی کنید.
- اطلاعات سرور و کلید را از مستندات معتبر آسان‌دسک دریافت کنید؛ تغییر این مقادیر به اطلاعات نامعتبر می‌تواند مانع اتصال شود.

منبع آموزش :وب‌سایت آسان‌دسک، بخش راهنمای استفاده و آموزش تنظیمات.

```
وب‌سایت: https://asandesk.ir/
```

نسخه ویندوز

```
https://asandesk.ir/download/win/
```

نسخه لینوکس

```
https://asandesk.ir/download/linux/
```

صفحه دانلود

```
https://asandesk.ir/download/
```

نرم‌افزارRustDesk ؛ کنترل از راه دور با امکان استفاده از سرور اختصاصی

معرفیRustDesk

RustDeskراست‌دسک یک نرم‌افزار کنترل از راه دور و متن‌باز است که امکان مشاهده و مدیریت رایانه‌ها و دستگاه‌های دیگر را از طریق شبکه فراهم می‌کند. این نرم‌افزار می‌تواند برای پشتیبانی فنی، مدیریت سرورها، عیب‌یابی سیستم‌ها و دسترسی به رایانه‌های راه دور مورد استفاده قرار گیرد.

یکی از قابلیت‌های مهمRustDesk ، امکان راه‌اندازی سرور اختصاصی برای مدیریت ارتباطات است. با استفاده از این قابلیت، سازمان‌ها و کاربران می‌توانند زیرساخت ارتباطی خود را مدیریت کنند و وابستگی به سرورهای عمومی نرم‌افزار را کاهش دهند.

RustDeskاز دو مؤلفه اصلی سرور استفاده می‌کند:

- **Hbbs**سرور شناسه و هماهنگ‌کننده ارتباط که در شناسایی دستگاه‌ها و برقراری ارتباط میان آن‌ها نقش دارد.
- **Hbbr**سرور واسط یا Relay که در صورت امکان‌پذیرنبودن اتصال مستقیم، ترافیک ارتباطی را میان دو دستگاه منتقل می‌کند.

RustDeskابتدا تلاش می‌کند ارتباط مستقیم همتابه‌همتا (P2P) برقرار کند. اگر به دلیل محدودیت‌های شبکه، NATیا فایروال این ارتباط ممکن نباشد، اتصال می‌تواند از طریق سرور Relay برقرار شود.

کاربردهایRustDesk

این نرم‌افزار برای سناریوهای مختلفی کاربرد دارد؛ از جمله:

- پشتیبانی از کاربران بدون نیاز به حضور فیزیکی در محل
- مدیریت رایانه‌های سازمانی و سرورهای داخلی
- عیب‌یابی سیستم‌عامل‌های Windows وLinux
- انتقال فایل میان دستگاه‌های متصل
- اشتراک‌گذاری کلیپ‌بورد و تبادل متن
- دسترسی به دستگاه‌های راه دور از طریق شبکه داخلی یا اینترنت
- راه‌اندازی زیرساخت کنترل از راه دور روی سرور اختصاصی

برای مثال، یک کارشناس فناوری اطلاعات می‌تواند با دریافت شناسه دستگاه و تأیید مجوز اتصال، به رایانه کاربر متصل شود و مشکل نرم‌افزاری آن را برطرف کند. همچنین، مدیر شبکه می‌تواند سرور اختصاصی RustDesk را راه‌اندازی کند تا مدیریت ارتباطات از طریق زیرساخت تحت کنترل سازمان انجام شود.

قابلیت‌های کلیدیRustDesk

۱. اتصال از راه دور

RustDeskامکان اتصال به دستگاه‌های دیگر را از طریق شناسه دستگاه و روش احراز هویت پشتیبانی‌شده فراهم می‌کند. بسته به تنظیمات نرم‌افزار، اتصال می‌تواند مستقیم یا از طریق سرور واسط انجام شود.

۲. پشتیبانی از سیستم‌عامل‌های مختلف

این نرم‌افزار برای سیستم‌عامل‌هایی مانندWindows ، macOS، Linux، Androidو iOS نسخه‌هایی دارد. قابلیت‌ها و محدودیت‌های هر نسخه ممکن است متفاوت باشد.

۳. سرور اختصاصی

با راه‌اندازی سرورهای hbbsو hbbrمی‌توان زیرساخت اختصاصی برای شناسایی دستگاه‌ها و انتقال ارتباط ایجاد کرد. در این حالت، آدرس سرور و کلید عمومی مربوط باید به‌درستی در کلاینت‌ها تنظیم شوند.

۴. انتقال فایل و کلیپ‌بورد

در جلسات پشتیبانی، امکان انتقال فایل و استفاده از کلیپ‌بورد می‌تواند تبادل اطلاعات و انجام عملیات فنی را ساده‌تر کند. دسترس‌پذیری این قابلیت‌ها به نسخه نرم‌افزار، سیستم‌عامل و مجوزهای اتصال بستگی دارد.

۵. تنظیم کیفیت تصویر

کیفیت تصویر و رزولوشن قابل تنظیم است تا بتوان میان وضوح تصویر و مصرف پهنای باند تعادل برقرار کرد. عملکرد واقعی به سخت‌افزار، کدک‌های قابل استفاده و شرایط شبکه بستگی دارد.

۶. پشتیبانی از چند نمایشگر

در سیستم‌ها و نسخه‌های سازگار، امکان مشاهده و مدیریت چند نمایشگر از راه دور وجود دارد. این قابلیت برای پشتیبانی از رایانه‌های کاری چندنمایشگری مفید است.

دانلود RustDesk برای ویندوز

در صفحه معرفی‌شده، فایل‌های زیر برای دانلود نسخه‌های ویندوز ارائه شده‌اند:

- **نسخه ۶۴ بیتی ویندوز:**

https://www.tisstsp.ir/wp-content/uploads/2026/04/RustDesk.64bit.rar

- **نسخه ۳۲ بیتی ویندوز:**

https://www.tisstsp.ir/wp-content/uploads/2026/04/RustDesk.32bit.rar

این لینک‌ها به فایل‌های فشرده RAR اشاره دارند. پیش از اجرا، فایل‌ها را از نظر اصالت و سلامت بررسی کنید. برای دریافت نسخه‌های دیگر، می‌توانید از مخزن رسمی پروژه RustDesk استفاده کنید:

https://github.com/rustdesk/rustdesk/releases

تنظیم RustDesk برای استفاده از سرور اختصاصی

برای استفاده از یک سرور اختصاصی، باید آدرس سرور شناسه، در صورت نیاز آدرس سرور Relay و کلید عمومی سرور را در تنظیمات شبکه نرم‌افزار وارد کنید.

مراحل کلی عبارت‌اند از:

۱. نرم‌افزار RustDesk را اجرا کنید.

۲. وارد بخش تنظیمات شوید.

۳. قسمت مربوط به شبکه یا سرور را باز کنید.

۴. آدرس سرور ID را در فیلد مربوط وارد کنید.

۵. در صورت نیاز، آدرس سرور Relay را تنظیم کنید.

۶. کلید عمومی سرور را در فیلد مربوط قرار دهید.

۷. تنظیمات را ذخیره کنید و اتصال را آزمایش کنید.

در صفحه منبع، دامنه desk.asiatech.irبه‌عنوان بخشی از تنظیمات معرفی شده است. با این حال، مقدار کامل تنظیمات و کلید عمومی باید از راهنمای همان سرویس دریافت شود؛ صرف واردکردن دامنه، جایگزین تنظیم کامل سرور اختصاصی نیست.

نکات امنیتی

- فایل نصب را ترجیحاً از مخزن رسمی پروژه یا منبع معتبر دریافت کنید.
- هنگام استفاده از سرور اختصاصی، کلید عمومی را از منبع قابل اعتماد دریافت کنید.
- برای دسترسی بدون حضور کاربر، رمز عبور قوی تعیین کنید و این قابلیت را فقط برای دستگاه‌های موردنیاز فعال نگه دارید.
- دسترسی به دستگاه‌های دیگر باید با مجوز مالک یا مدیر مسئول انجام شود.

- استفاده از سرور اختصاصی به‌تنهایی تضمین نمی‌کند که همه داده‌ها در تمام شرایط فقط از مسیر داخلی عبور کنند؛ مسیر واقعی ارتباط به تنظیمات و وضعیت شبکه بستگی دارد.

LEARNING HIVE

contact me:

seilany.ir
learninghive.ir
emperor-os.ir
emperor-os.com
predator-os.com
telegram:@seilany

HosseinSeilani

Designer,Developer and Linux sysadmin



Founder and Developer of Emperor-OS, Little-Psycho and Predator-OS. I bring significant experience as a Linux/Windows sysadmin and graphical web design,UX/UI to the Open Source Community.

LITTLE-PSYCHO
DESTRUCTIVE LINUX

Emperor-OS

Predator-OS