

# Serveur de cache APT

Script d'auto configuration d'un serveur de cache pour APT, côté serveur et client.

*Apt-Cacher NG est un mandataire de cache pour le téléchargement de paquets depuis des dépôts de logiciels dans le style de Debian (ou d'autres types).. Le principe est qu'une machine centrale héberge le mandataire pour un réseau local. Les clients règlent leur configuration d'APT pour télécharger sur cette machine. Apt-Cacher NG conserve une copie de toutes les données utiles transitant à travers lui et, quand une requête similaire est faite, la copie en cache des données est délivrée sans être téléchargée à nouveau.*

Ci dessus la version 0.1.0

Mais préférable de suivre la version du GIT : <https://github.com/CyrilleBiot/scripts/blob/master/acn.py>

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

""" Script d'installation et de configuration du serveur de cache apt
    apt-cacher-ng soit en tant que serveur (ajout du paquet sur le système
    soit en tant que client (creation d'un fichier de proxy apt)
    """

__author__ = "Cyrille BIOT"
__copyright__ = "Copyleft"
__credits__ = "Cyrille BIOT"
__license__ = "GPL"
__version__ = "0.1.2"
__date__ = "2020/01/26"
__maintainer__ = "Cyrille BIOT"
__email__ = "cyrille@cbiot.fr"
__status__ = "Devel"

import os, re, sys
import platform, subprocess, socket

def baseDebian():
    """ Fonction permettant de connaitre le Systeme d'exploitant faisant
    tourner le script
    Ou DEBIAN ou UBUNTU pour savoir si on utilise su ou sudo
    Retourne une variable de type string (admin)
    """
    # Ubuntu ou DEBIAN
    if 'Debian' in platform.version():
        # Si DEBIAN, verif si root lance le script
        admin = 'su'
        print('Vous utilisez un système Debian (su pour administration).')
        if not os.geteuid() == 0:
            sys.exit("Seul le root peut lancer ce script. Nécessite
```

```
privilèges administrateur")
    else:
        print('Vous utilisez un système non Debian (sudo pour
adminstration).')
        admin = 'sudo'
    return admin

def installServeur():
    """ Fonction installant le serveur de cache apt-cacher-ng """
    portACN = 3142
    ipServeur = ipRecuperation()

    # Installation SERVEUR
    # Tester si le package apt-cacher-ng est installé ou non
    retval = subprocess.call(["which", "apt-cacher-ng"])
    if retval != 0:
        print("Le package apt-cacher-ng n'est pas installé. Installation...")

        # Paramètres de l'install
        cmdInstall = ['apt-get', 'install', 'apt-cacher-ng', '-y']
        cmdUpdate = ['apt-get', 'update']
        # Debian, Ubuntu
        admin = baseDebian()

        # Adaptation système Ubuntu
        if admin == 'sudo':
            cmdInstall.insert(0, 'sudo')
            cmdUpdate.insert(0, 'sudo')

        # On installe le paquet
        subprocess.run(cmdInstall)
        subprocess.run(cmdUpdate)
    else:
        print('Le package apt-cacher-ng est déjà présent sur votre
système.')
        sys.exit()

    # Affichage Informations
    print("=====")
    print("Le serveur de cache est dès lors opérationnel")
    print("Le port d'écoute est : {}".format(portACN))
    print("Page d'aministration : http://{}/acng-
report.html".format(ipServeur, portACN))
    print("Notez bien l'ip de votre serveur, elle vous sera indispensable
pour la configuration des clients.")
    print("L'IP du serveur est : {}".format(ipServeur))
    print("Indispensable : cette IP doit être FIXE (réglage sur votre BOX ou
serveur DHCP).")
```

```

def installClient(ip):
    """ Fonction installant un fichier de configuration apt pour les postes
    clients
        Créer un fichier dans /etc/apt/apt.conf.d/ ayant pour nom
    00aptproxyANC
    """
    print("Installation client.")
    # Reste à insérer l'ip et à le coller au bon endroit
    msgApt = 'Acquire::http::Proxy "' + ip + ':3142";\n'
    dirInstall = '/etc/apt/apt.conf.d/'
    fileName = '00aptproxyANC'
    fileLocInstall = dirInstall + fileName
    fichier = open(fileLocInstall, "w")
    fichier.write(msgApt)
    fichier.close()

def ipRecuperation():
    """ Fonction récupérant l'adresse IPv 4 de la machine"""
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
    s.connect(('<broadcast>', 0))
    return s.getsockname()[0]

def ipTest(ip):
    """ Fonction testant la validité d'une adresse IPv4 """
    reg =
r"^([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|
1[0-9]{2}|2[0-4][0-9]|25[0-5])$"
    if re.match(reg, ip):
        return True
    else:
        return False

# =====
# Lancement du script
# =====
def clientServeur():
    """ Fonction déterminant s'il s'agit d'une installation de type Serveur
    ou Client
        Retourne une variable string
    """
    while True:
        try:
            choixInstall = input("Type d'installation (client/serveur) : ")
            if choixInstall.lower() in ['client', 'serveur']:
                print('Installation de type {}'.format(choixInstall))
                break
            else:

```

```
        print('Préciser : client OU serveur.')
        print('ATTENTION A LA CASSE. Pas de majuscule.')
    except ValueError:
        print("Oops!  Réponse incorrecte... Réessayer...")
    return choixInstall
```

*# Procédure d'installation*

```
choixInstall = clientServeur()
if choixInstall.lower() == 'serveur':
    installServeur()
else:
    # Installation client
    while True:
        try:
            ipServeur = input("Saisir l'IP du Serveur :")
            if ipTest(ipServeur) is True:
                break
        except ValueError:
            print("Oops!  Réponse incorrecte... Réessayer...")
    installClient(ipServeur)
```

## Lien

- [Le paquet sous DEBIAN SID](#)
- [Homepage Apt-cacher-ng](#)

From:  
<https://cbiot.fr/dokuwiki/> - **Cyrille BIOT**

Permanent link:  
<https://cbiot.fr/dokuwiki/python:acn-py-installer?rev=1580050986>

Last update: **2020/01/26 15:03**

