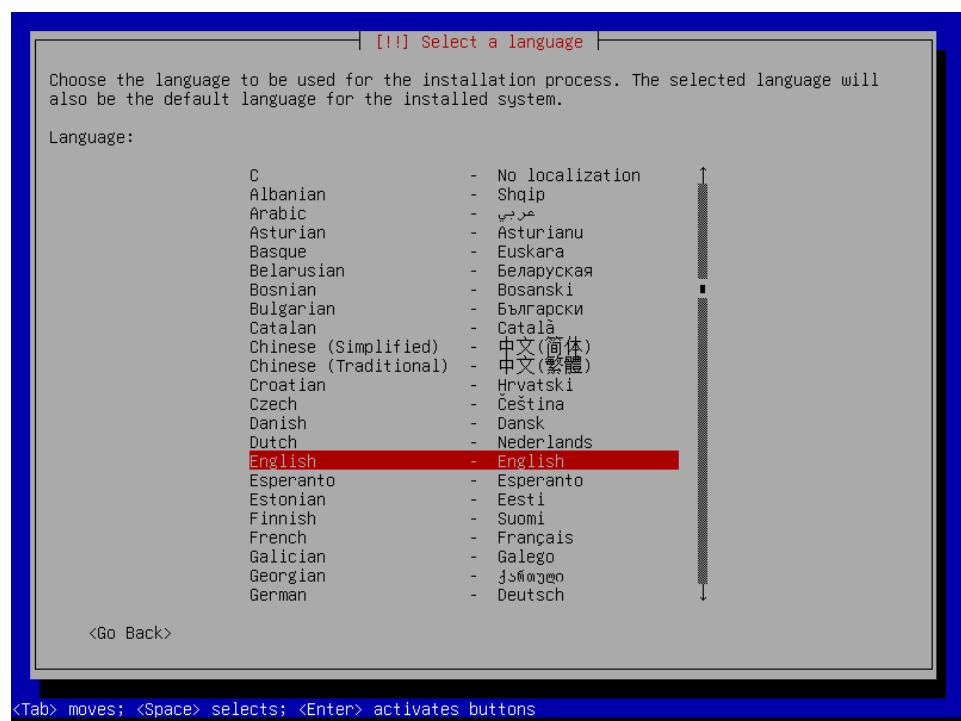
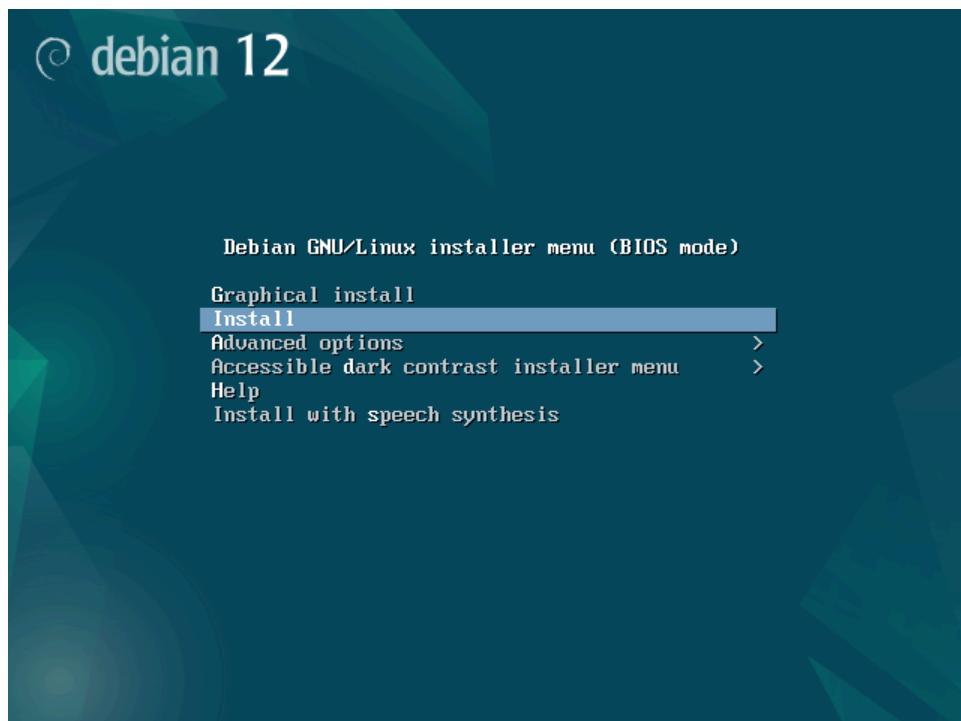


# INSTALLATION DU SERVEUR VOIP

## I - MISE EN PLACE D'UNE VM DEBIAN (AVEC OU SANS GUI)

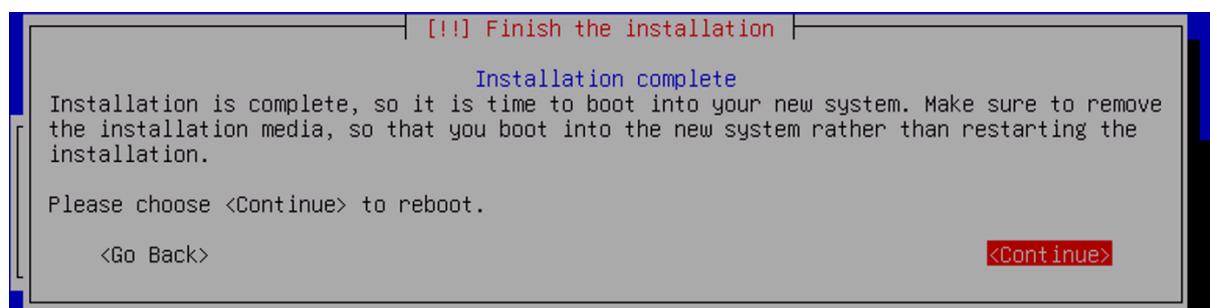
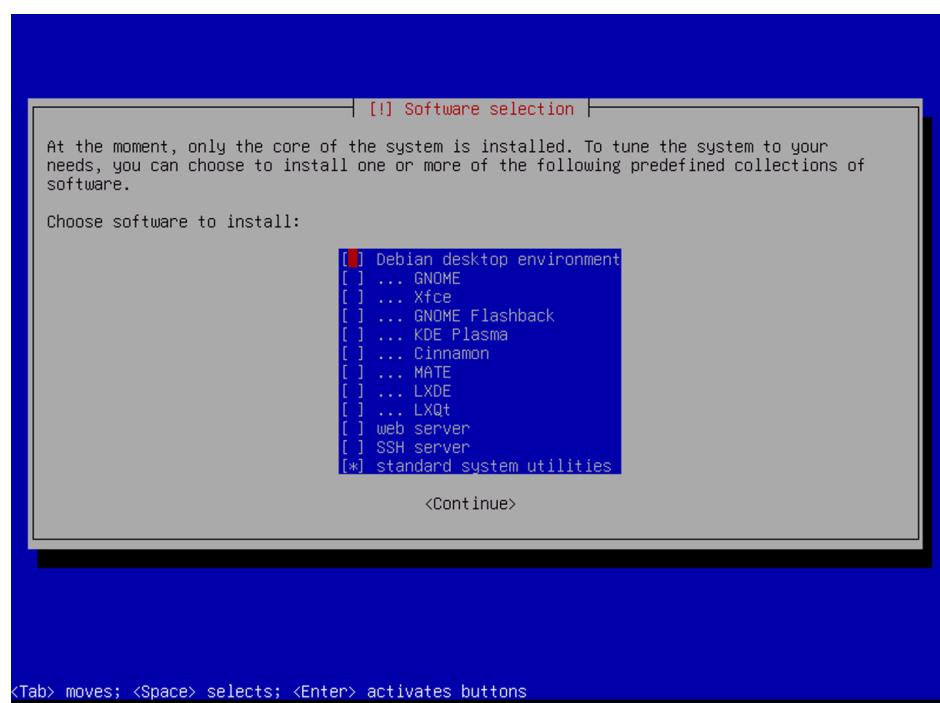
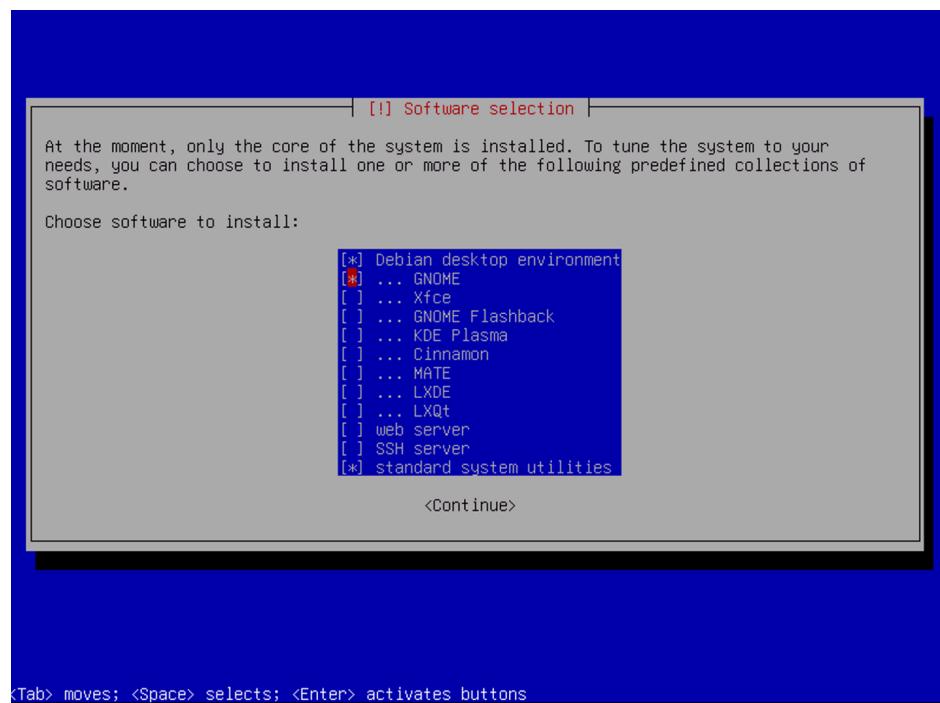


[!] Configure the network	
Please enter the hostname for this system.	
The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.	
Hostname:	
<input type="text" value="voipserver"/>	
<a href="#">&lt;Go Back&gt;</a>	<a href="#">Continue&gt;</a>

[!] Set up users and passwords	
You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.	
A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.	
The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.	
Note that you will not be able to see the password as you type it.	
Root password:	
<input type="password" value="****"/>	
<input type="checkbox"/> Show Password in Clear	
<a href="#">&lt;Go Back&gt;</a>	<a href="#">Continue&gt;</a>

[!] Set up users and passwords	
A user account will be created for you to use instead of the root account for non-administrative activities.	
Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.	
Full name for the new user:	
<input type="text" value="VoIP Server"/>	
<a href="#">&lt;Go Back&gt;</a>	<a href="#">Continue&gt;</a>

[!] Set up users and passwords	
Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.	
Username for your account:	
<input type="text" value="voip"/>	
<a href="#">&lt;Go Back&gt;</a>	<a href="#">Continue&gt;</a>



## II - PRÉPARATION DE L'ENVIRONNEMENT

- Après avoir installé Debian, on met à jour le système d'exploitation.  
(les trucs basiques)

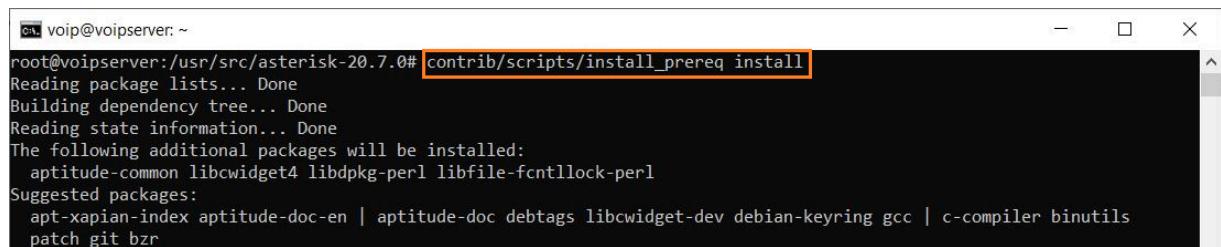
```
1 - apt update  
2 - apt upgrade
```

- Maintenant, nous allons télécharger **Asterisk** et le décompresser.

```
1 - cd /usr/src  
2 - wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20-current.tar.gz  
3 - tar xvf asterisk-20-current.tar.gz
```

- À l'étape suivante, nous allons préparer l'environnement pour compiler et exécuter le serveur Asterisk , Le script **install\_prereq** identifie et installe les dépendances système, et autres composants nécessaires pour l'installation et au bon fonctionnement d'Asterisk.

```
1 - contrib/scripts/install_prereq install
```



```
voip@voipserver: ~  
root@voipserver:/usr/src/asterisk-20.7.0# contrib/scripts/install_prereq install  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  aptitude-common libcwidget4 libdpkg-perl libfile-fcntllock-perl  
Suggested packages:  
  apt-xapian-index aptitude-doc-en | aptitude-doc debtags libcwidget-dev debian-keyring gcc | c-compiler binutils  
  patch git bzr
```

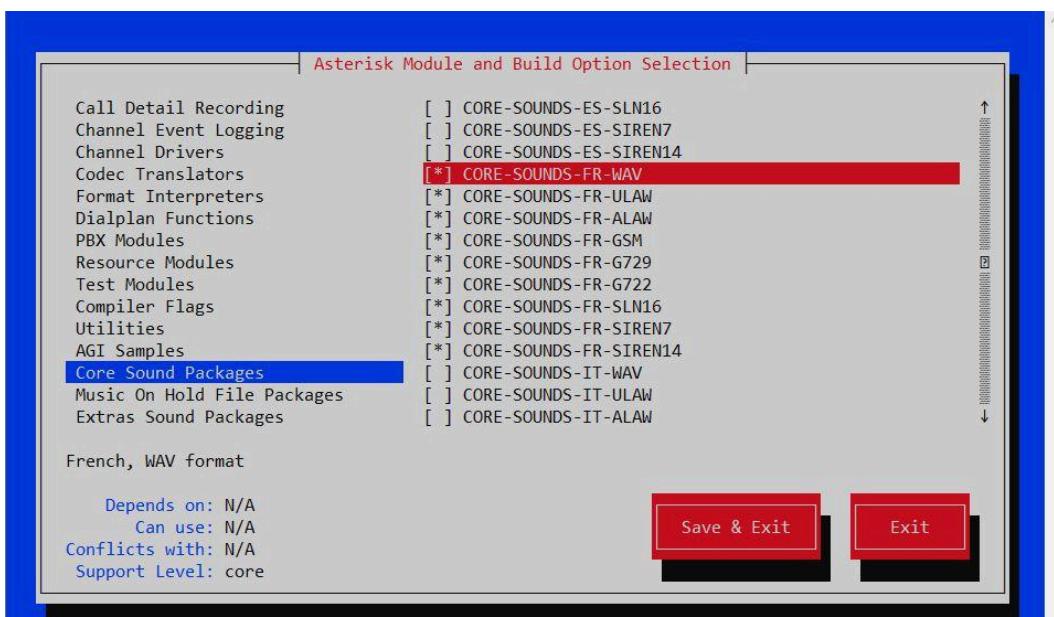
### **III – INSTALLATION D'ASTERISK**

Ensuite, on va exécuter la commande `./configure`, qui effectuera plusieurs vérifications sur le système et préparera Asterisk à être installé sur le serveur.

1 - *./configure*

Après avoir réussi à exécuter [./configure](#), vous devriez voir un message similaire à celui-ci :

**make menuselect** (optionnel, ici pour ajouter les packs fr ou autre )



- make
- make install
- make samples
- make config
- systemctl start asterisk
- sudo asterisk -rv (apt install sudo, seulement si ça ne veut pas se lancer)
- exit (ou ctrl+c)
- cd /etc/asterisk

## IV – CRÉATION ET CONFIGURATION DE COMPTES SIP

### Avant de commencer :

1. **Sauvegarde du fichier existant** : Renommez le fichier **pjsip.conf** actuel en **pjsip.conf.old** pour conserver une copie de sauvegarde.

```
root@voipserver:/etc/asterisk# ls | grep pjsip.conf
pjsip.conf
root@voipserver:/etc/asterisk# mv pjsip.conf pjsip.conf.old
root@voipserver:/etc/asterisk# ls | grep pjsip.conf
pjsip.conf.old
root@voipserver:/etc/asterisk#
```

2. **Création d'un nouveau fichier** : Créez un nouveau fichier **pjsip.conf** vierge en utilisant la commande **nano pjsip.conf**.

```
root@voipserver:/etc/asterisk# ls | grep pjsip.conf
pjsip.conf
pjsip.conf.old
root@voipserver:/etc/asterisk#
```

## Création et configuration des comptes SIP :

1. Ouvrir le fichier **pjsip.conf** nouvellement créé avec nano
2. Ajouter les configurations des comptes SIP : Insérez les informations de configuration spécifiques à chaque compte SIP dans le fichier **pjsip.conf**.

Ces informations incluent généralement :

Config basic :

```
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[endpoint_cowork]
type=endpoint
context=cowork
disallow=all
allow=ulaw
allow= h263
language=fr

[auth_userpass]
type=auth
auth_type=userpass

[aor_dynamic]
type=aor
max_contacts=1

[allaoui](endpoint_cowork)
auth=allaoui
aors=allaoui

[allaoui](auth_userpass)
password=1234
username=allaoui

[allaoui](aor_dynamic)
```

Secure des appels/media avec **TLS et SDES**

```
[transport-tls]
type=transport
protocol=tls
bind=0.0.0.0:5061
cert_file=/etc/asterisk/keys/asterisk.crt
priv_key_file=/etc/asterisk/keys/asterisk.key
method=sslv23

[endpoint_cowork]
type=endpoint
context=cowork
disallow=all
allow=ulaw
media_encryption=sdes
language=fr
```

```
voip@voipserver: ~
GNU nano 7.2          pjsip.conf

[transport-tls]
type=transport
protocol=tls
bind=0.0.0.0:5061
cert_file=/etc/asterisk/keys/asterisk.crt
priv_key_file=/etc/asterisk/keys/asterisk.key
method=sslv23

[endpoint_cowork]
type=endpoint
context=cowork
disallow=all
allow=ulaw
media_encryption=sdes
language=fr

[auth_userpass]
type=auth
auth_type=userpass

[aor_dynamic]
type=aor
max_contacts=1
```

## Sécurisation des appels :

1. On va créer un dossier keys dans le rep **/etc/asterisk** ou vont être logés nos clés : `mkdir /etc/asterisk/keys`
2. Ensuite, on utilisez le script "**ast\_tls\_cert**" dans le répertoire source d'asterisk "**contrib/scripts**" qui va nous permettre de créer une autorité de certification auto-signée et un certificat Asterisk

```
./ast_tls_cert -C 192.168.0.35 -O "Server VoIP" -d /etc/asterisk/keys -b 2048
```

```
root@voipserver:/usr/src/asterisk-20.7.0/contrib/scripts# ./ast_tls_cert -C 192.168.0.35 -O "Server VoIP" -d /etc/asterisk/keys -b 2048
No config file specified, creating '/etc/asterisk/keys/tmp.cfg'
You can use this config file to create additional certs without
re-entering the information for the fields in the certificate
Creating CA key /etc/asterisk/keys/ca.key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
Creating CA certificate /etc/asterisk/keys/ca.crt
Enter pass phrase for /etc/asterisk/keys/ca.key:
Creating certificate /etc/asterisk/keys/asterisk.key
Creating signing request /etc/asterisk/keys/asterisk.csr
Creating certificate /etc/asterisk/keys/asterisk.crt
Certificate request self-signature ok
subject=CN = 192.168.0.35, O = Server VoIP
Enter pass phrase for /etc/asterisk/keys/ca.key:
Combining key and crt into /etc/asterisk/keys/asterisk.pem
```

L'option "**-C**" est utilisée pour définir notre hôte - nom **DNS** ou l'**adresse IP**.

L'option "**-O**" définit notre nom d'entité.

L'option "**-d**" est le répertoire de sortie des clés.

L'option "**-b**" spécifie la taille du fichier de clé privée, par défaut 1024.

Ensuite on sera invité à saisir un mot de passe pour :

`/etc/asterisk/keys/ca.key`, Cela créera le fichier `/etc/asterisk/keys/ca.crt`.

De même pour ici `/etc/asterisk/keys/asterisk.key` +

`/etc/asterisk/keys/asterisk.crt` sera automatiquement généré.

`/etc/asterisk/keys/asterisk.pem` sera créé, une combinaison des fichiers `asterisk.key` et `asterisk.crt`.

```
root@voipserver:/etc/asterisk/keys# root@voipserver:/etc/asterisk/keys# ls
asterisk.crt  asterisk.csr  asterisk.key  asterisk.pem  ca.cfg  ca.crt  ca.key  tmp.cfg
root@voipserver:/etc/asterisk/keys#
```

## Mise en place du context des appels et de la messagerie vocal

- cd /etc/asterisk
- nano extensions.conf

```
root@voipserver:/etc/asterisk# ls | grep extensions.conf
extensions.conf
extensions.conf.old
root@voipserver:/etc/asterisk#
```

---

### [cowork]

```
exten => 6001,1,Answer()
exten => 6001,2,Dial(PJSIP/allaoui,30) ; Après 30 secondes envoi vers la règle 3
exten => 6001,3,Playback(vm-nobodyavail)
exten => 6001,4,VoiceMail(6001) ; répondeur compte
exten => 6001,5,Hangup()
exten => 6099,1,VoiceMailMain() ; 6099 Numéro de téléphone du répondeur
```

---

- nano voicemail.conf

```
root@voipserver:/etc/asterisk# ls | grep voicemail.conf
voicemail.conf
voicemail.conf.old
root@voipserver:/etc/asterisk#
```

---

### [general]

```
format = wav49|gsm|wav|ulaw
```

### [default]

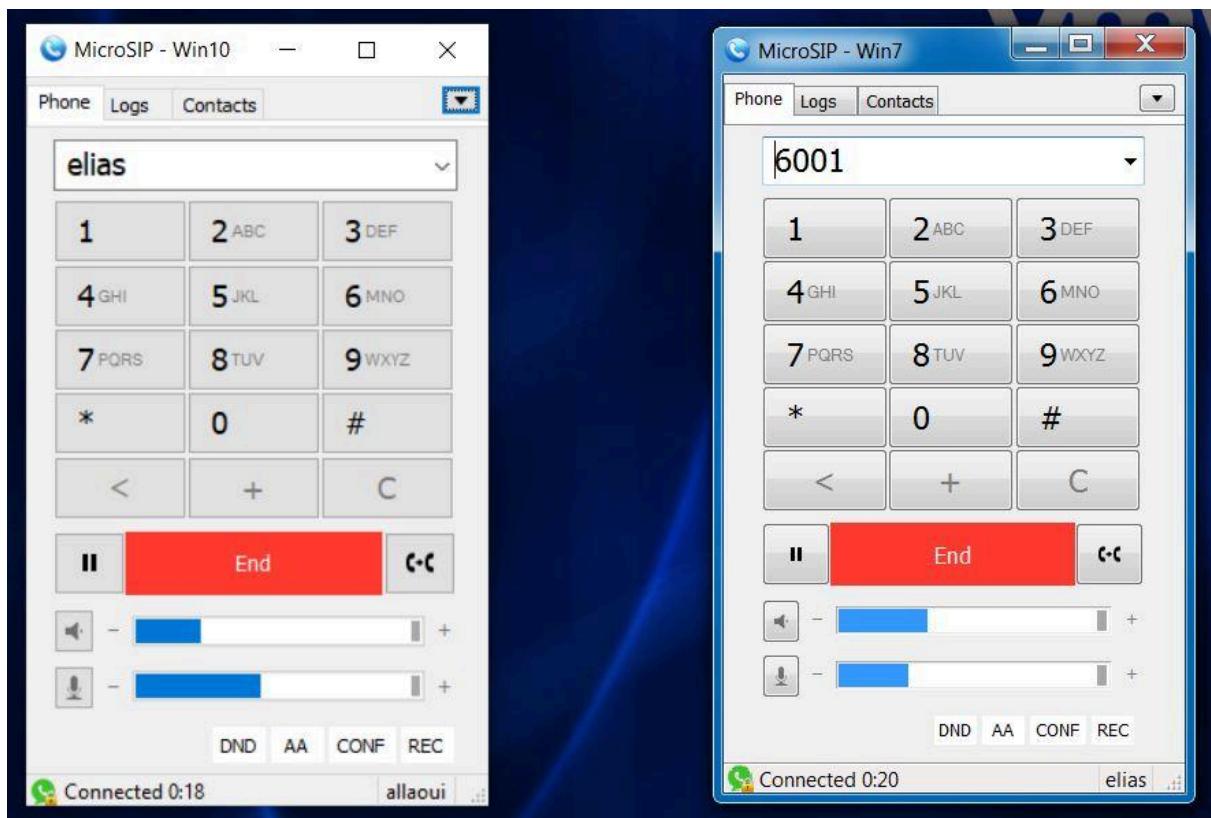
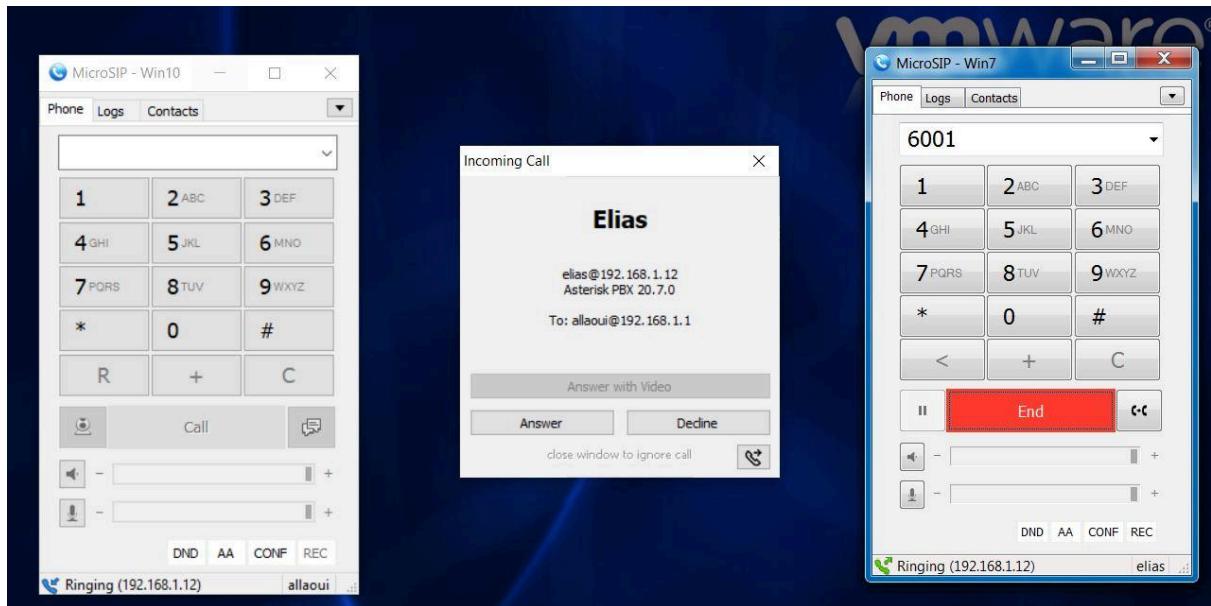
```
6001 => 1234, allaoui
6002 => 1234, elias
6003 => 1234, rachid
```

---

1234 = mot de passe d'accès à la messagerie, nom d'utilisateur

- sudo asterisk -rx "core restart now" (Charger la configuration)
- sudo asterisk -rv
- pjsip list aors (lister les clients enregistrés)

## TEST ET VALIDATION (UPDATE A FAIRE !!!!)



## Fixer une adresse IP au serveur

Pour connaître l'interface réseau, utilisez la commande "ip a", par exemple : `ens33`.

```
root@voipserver:/home/voip# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:5f:41:bb brd ff:ff:ff:ff:ff:ff
        altname enp2s1
```

Utilisez "ip route show" pour afficher la passerelle par défaut actuellement utilisée par notre machine.

Pour connaître notre nameserver, utilisez "cat /etc/resolv.conf".

Ensuite, accédez à `/etc/network/interfaces` et appliquez la configuration suivante:

- Remplacez "inet dhcp" par "static".
- Attribuez-lui l'**adresse ip** que vous avez choisie, ainsi que son **masque réseau** et enfin la **passerelle**."

```
# The primary network interface
allow-hotplug ens33
iface ens33 inet static
    address 192.168.1.12/24
    netmask 255.255.255.0
    gateway 192.168.1.2
    dns-nameservers 192.168.1.2
    broadcast 192.168.1.255
```

Pour ajout d'un second adaptateur :

1. sudo apt-get install net-tools

```
# The primary network interface
allow-hotplug ens33
iface ens33 inet static
    address 192.168.1.12/24
    netmask 255.255.255.0
    gateway 192.168.1.2
    dns-nameservers 192.168.1.2
    broadcast 192.168.1.255

    allow-hotplug ens37
    iface ens37 inet dhcp
```

- 2.

```
root@voipserver:/home/voip# cat /etc/resolv.conf
domain laplateforme.io
search laplateforme.io
nameserver 192.168.1.2
nameserver 10.10.0.1
root@voipserver:/home/voip# sudo chattr +i /etc/resolv.conf
root@voipserver:/home/voip# systemctl restart networking
```

- 3.

## V – MISE EN PLACE D’UN AUTOMATE

Pour mettre en place notre automate, nous aurons besoin d'un système appelé **IVR**.

### Qu'est-ce qu'un IVR ?

Un **IVR (Interactive Voice Response)** guide les utilisateurs vers le service qu'ils veulent contacter grâce à des options de menu prédéfinies.

j

En choisissant la bonne option, les appels sont dirigés vers le département, l'individu ou la file d'attente correspondant.

À l'intérieur d'un IVR, vous pouvez configurer des règles spéciales pour déterminer qui prend chaque appel en fonction des compétences, du volume d'appels ou de la période de temps.

---

### First Step !

La mise en place d'un IVR se fait en deux étapes : d'abord, on apprend à Asterisk à parler, puis on crée l'IVR.

### Comment faire parler Asterisk ?

Pour faire parler Asterisk, on installe un outil de synthèse vocale appelé **GoogleTTS**.

### Installation de GoogleTTS

D'abord, GoogleTTS a besoin de quelques éléments pour fonctionner :

1. Installer les dépendances : `sudo apt-get install perl libwww-perl sox mpg123`
2. Télécharger le script GoogleTTS :
  - `cd /var/lib/asterisk/agi-bin` puis
  - `wget https://raw.github.com/zaf/asterisk-googleTTS/master/googleTTS.agi`
3. Donner les permissions nécessaires : `chmod +x googleTTS.agi`

**Maintenant, GoogleTTS est installé. Voyons comment l'utiliser.**

## Configuration des IVR

La configuration des IVR se fait dans le fichier `extensions.conf`

1. Accéder au fichier : `cd /etc/asterisk` puis ouvrir `extensions.conf` avec l'éditeur de texte nano.
2. Ajouter le contexte souhaité (par exemple, "`menu`")

---

`exten => 100,1,Goto(menu,s,1) ; 100 = numéro pour accéder au menu (vous pouvez en choisir un autre)`

### [menu]

`exten => s,1,Answer()`

`exten => s,2,agi(googletts.agi,"Bonjour !!!! Cela fonctionne, une voix me parle !!!",fr)`

`exten => s,3,Hangup()`

---

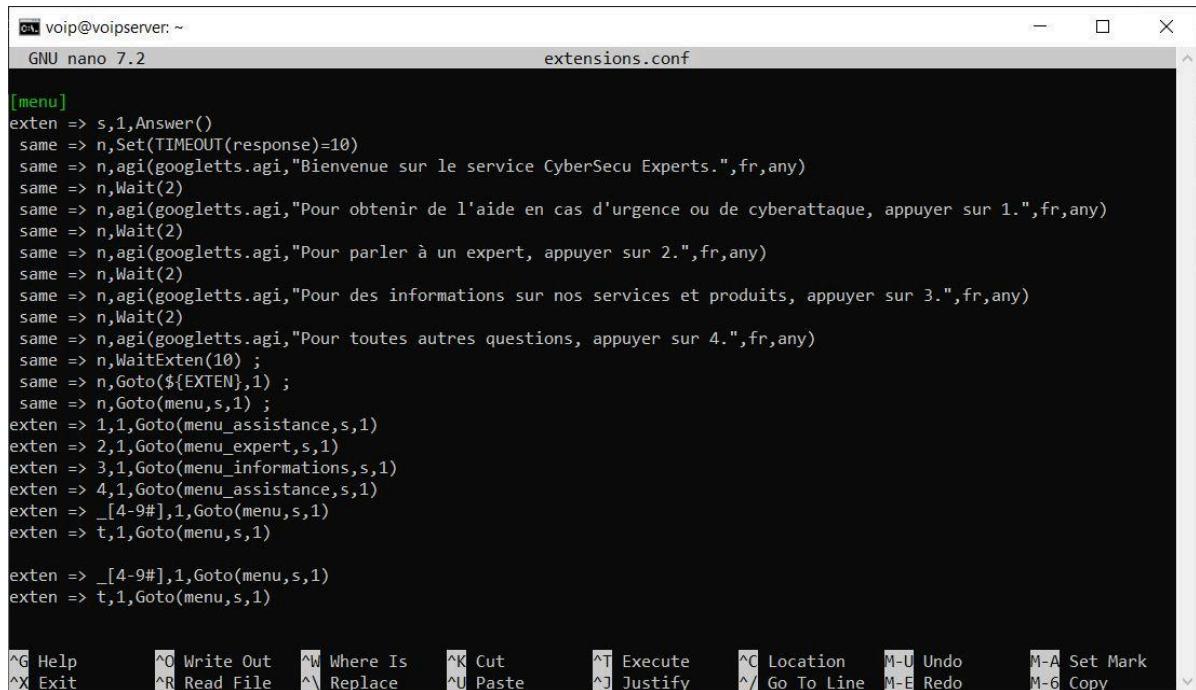
**Maintenant, vous pouvez restart la configuration et tester votre IVR.**

- `sudo asterisk -rx "core restart now"`

Le reste, c'est à vous ! Vous pouvez modifier le menu comme vous le souhaitez.

L'essentiel est fait : **ASTERISK PARLE !**

Voici le menu que j'ai fait pour m'amuser



```
voip@voipserver: ~
GNU nano 7.2                                         extensions.conf

[menu]
exten => s,1,Answer()
same => n,Set(TIMEOUT(response)=10)
same => n,agi(googletts.agi,"Bienvenue sur le service CyberSecu Experts.",fr,any)
same => n,Wait(2)
same => n,agi(googletts.agi,"Pour obtenir de l'aide en cas d'urgence ou de cyberattaque, appuyer sur 1.",fr,any)
same => n,Wait(2)
same => n,agi(googletts.agi,"Pour parler à un expert, appuyer sur 2.",fr,any)
same => n,Wait(2)
same => n,agi(googletts.agi,"Pour des informations sur nos services et produits, appuyer sur 3.",fr,any)
same => n,Wait(2)
same => n,agi(googletts.agi,"Pour toutes autres questions, appuyer sur 4.",fr,any)
same => n,WaitExten(10) ;
same => n,Goto(${EXTEN},1) ;
same => n,Goto(menu,s,1) ;
exten => 1,1,Goto(menu_assistance,s,1)
exten => 2,1,Goto(menu_expert,s,1)
exten => 3,1,Goto(menu_informations,s,1)
exten => 4,1,Goto(menu_assistance,s,1)
exten => _[4-9#],1,Goto(menu,s,1)
exten => t,1,Goto(menu,s,1)

exten => _[4-9#],1,Goto(menu,s,1)
exten => t,1,Goto(menu,s,1)

^G Help      ^O Write Out   ^W Where Is    ^K Cut      ^T Execute    ^C Location   M-U Undo    M-A Set Mark
^X Exit     ^R Read File   ^\ Replace    ^U Paste    ^J Justify   ^/ Go To Line M-E Redo    M-6 Copy
```

## Mise en place d'une solution d'appels automatisés pour la prospection

Premièrement avant de mettre en place notre solution on va d'abord vérifier une chose, dans le dossier **etc/asterisk**.

On vérifie dans le fichier **modules.conf** pour voir si un truc est bien activé (**autoload=yes**).

Ensuite, on s'assure que le fichier **pbx\_spool.so** est bien chargé (**load = pbx\_spool.so**). Si ça n'est pas le cas, ça marchera pas.

```
GNU nano 7.2      modules.conf *
;
; Asterisk configuration file
;
; Module Loader configuration file
;

[modules]
autoload=yes

load = pbx_spool.so
```

Pour créer notre liste de contacts, on va utiliser un format simple, un fichier txt qu'on va appeler **liste\_numeros.txt**

On va écrire les numéros des contacts qu'on va prospector à l'intérieur.

```
root@voipserver:/usr/src# cat liste_numeros.txt
allaoui
elias
rachid
root@voipserver:/usr/src#
```

**PS :** Dans Asterisk, il y a un dossier spécial (**/var/spool/asterisk/outgoing**) où si on dépose un fichier, Asterisk le détecte automatiquement et l'utilise pour passer des appels. C'est là-dedans qu'on va placer nos contacts.

Pour le faire, on va utiliser un script Python que l'on va configurer.

## script.py

```
root@voipserver:/usr/src# cat script.py
import random
import os

# Lire la liste de numéros depuis le fichier texte
with open('liste_numeros.txt', 'r') as file:
    contacts = file.read().splitlines()

# Randomiser la liste des contacts
random.shuffle(contacts)

# Chemin du répertoire sortant d'Asterisk
outgoing_dir = '/var/spool/asterisk/outgoing/'

# Créer des fichiers d'appel Asterisk pour chaque contact
for i, contact in enumerate(contacts, start=1):
    filename = os.path.join(outgoing_dir, f'appel_{i}.call')
    with open(filename, 'w') as call_file:
        call_file.write(f'Channel: PJSIP/{contact}\n')
        call_file.write('MaxRetries: 3\n')
        call_file.write('RetryTime: 60\n')
        call_file.write('WaitTime: 30\n')
        call_file.write('Context: prospection\n')
        call_file.write('Extension: s\n')
        call_file.write('Priority: 1\n')
root@voipserver:/usr/src#
```

## Fichier généré à la sortie par notre script :

```
root@voipserver:/var/spool/asterisk/outgoing# ls
appel_1.call  appel_2.call  appel_3.call
root@voipserver:/var/spool/asterisk/outgoing#
```

## Contenu d'un fichier :

```
GNU nano 7.2
Channel: PJSIP/allaoui
MaxRetries: 3
RetryTime: 60
WaitTime: 30
Context: prospection
Extension: s
Priority: 1
```

## Context dans extensions.conf

```
[prospection]
exten => s,1,AGI(googletts.agi, "Bienvenue chez CyberSecu Expert")
        same => n,AGI(googletts.agi, "Avec notre offre de base, nous o")
        same => n,AGI(googletts.agi, "En plus de notre offre de base, ")
        same => n,AGI(googletts.agi, "Si vous êtes intéressé par nos s")
        same => n,Hangup()
```

**Channel** = Ce paramètre spécifie le canal sur lequel Asterisk doit placer l'appel sortant.

**MaxRetries** = Ce paramètre spécifie le nombre maximum de tentatives d'appel en cas d'échec. Si l'appel initial échoue, Asterisk tentera de réessayer jusqu'à MaxRetries fois avant d'abandonner.

Par exemple, **MaxRetries: 3**, Asterisk tentera l'appel jusqu'à trois fois au total.

**RetryTime** = Ce paramètre spécifie le délai (en secondes) entre chaque tentative d'appel en cas d'échec. Après un échec d'appel, Asterisk attendra le nombre de secondes spécifié dans RetryTime avant de retenter l'appel.

Par exemple, **RetryTime: 60**, Asterisk attendra 60 secondes avant de retenter l'appel après un échec.

**WaitTime** = Ce paramètre spécifie le temps maximum (en secondes) à attendre pour que l'appel soit connecté avant de passer à l'étape suivante ou d'abandonner. Si l'appel ne se connecte pas dans le délai spécifié par WaitTime, Asterisk considérera l'appel comme un échec.

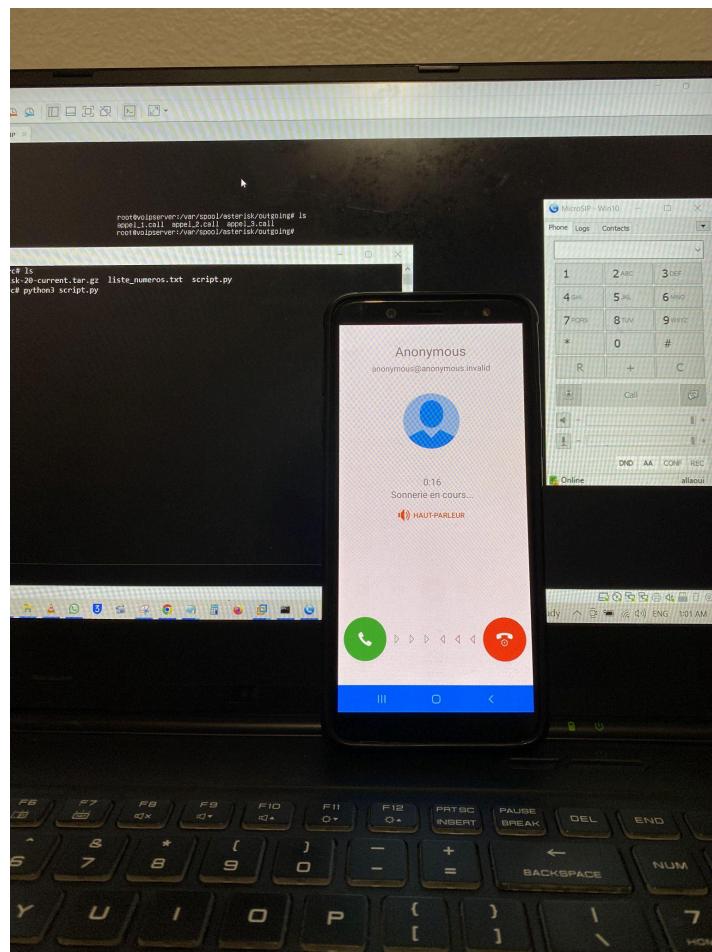
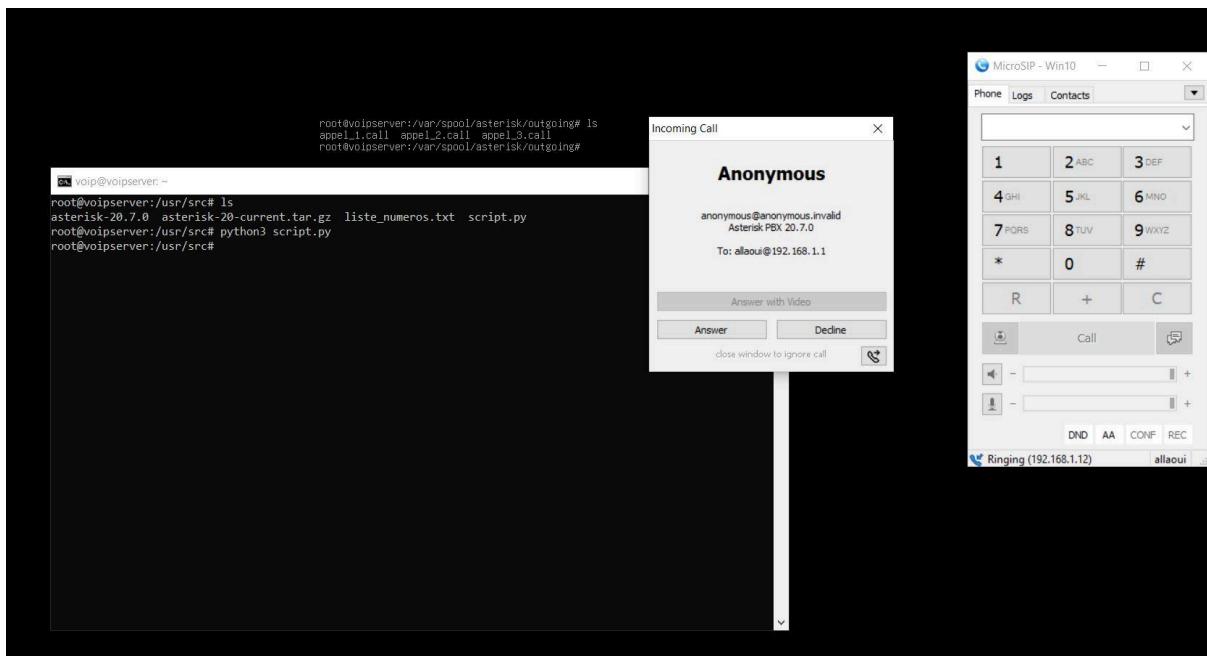
Par exemple, ici **WaitTime: 30**, Asterisk attendra jusqu'à 30 secondes pour que l'appel soit connecté.

**Context** = Ce paramètre spécifie le contexte dans lequel les appels sortants doivent être traités.

Le contexte détermine **les règles de routage** et **les actions à prendre** pour les appels sortants.

**Extension** = Ce paramètre spécifie l'extension dans le contexte spécifié à laquelle Asterisk doit envoyer l'appel sortant.

**Priority** = Ce paramètre spécifie la priorité dans le contexte et l'extension spécifiés à laquelle Asterisk doit envoyer l'appel sortant.  
Si plusieurs actions sont définies pour une extension donnée, la priorité détermine l'ordre dans lequel elles doivent être exécutées.



## VI – DÉPLOIEMENT AUTOMATISÉ

### Écrire un script permettant l'installation d'Asterisk sur une nouvelle VM .

Une fois que vous avez terminé l'installation de votre nouvelle machine virtuelle, il vous suffit de suivre ces étapes simples pour installer Asterisk :

1. Créez un fichier nommé selon votre préférence, par exemple, utilisez la commande "**nano script.sh**".
2. Copiez et collez le script dans ce fichier, puis enregistrez-le.
3. Modifiez les autorisations du fichier pour lui permettre d'être exécuté en utilisant la commande "**chmod +x script.sh**".
4. Enfin, lancez le script en exécutant la commande "**./script.sh**".

```
root@debian:/home/test# nano script_onthefly.sh
root@debian:/home/test# chmod +x script_onthefly.sh
root@debian:/home/test# ./script_onthefly.sh
```

===== > script\_onthefly.sh <=====

```
test@debian: ~
GNU nano 7.2                                     script_onthefly.sh *
#!/bin/bash

# Mettre à jour les paquets système
apt update && sudo apt upgrade -y

#On install sudo
apt-get install sudo -y

# Télécharger et installer Asterisk
cd /usr/src
wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20-current.tar.gz
tar xvzf asterisk-20-current.tar.gz
cd asterisk-20.7.0/

# Configurer Asterisk
./contrib/scripts/install_prereq install
./configure
make
make install
make samples
make config

# Activer et démarrer Asterisk
sudo systemctl enable asterisk
sudo systemctl start asterisk
```

Video de démonstration

[FormatFactory Screen Record20240505\\_230405.mp4 – Google Drive](#)

---

## VII – SÉCURISATION

Nous allons installer un système de supervision pour surveiller notre serveur, et pour cela, nous utiliserons **Zabbix**.

**Zabbix** est une solution logicielle qui permet de surveiller une multitude de paramètres réseau ainsi que la santé et l'intégrité des serveurs, des machines virtuelles, des applications, des services, des bases de données, des sites web, et même des environnements cloud.

Grâce à son mécanisme de notification flexible, les utilisateurs peuvent configurer des alertes par e-mail pour pratiquement tous les événements, assurant ainsi une réactivité rapide aux problèmes serveurs.

De plus, Zabbix propose des fonctionnalités avancées de reporting et de visualisation des données, basées sur les données collectées, ce qui en fait un outil idéal pour la gestion de la capacité.

---

Pour ce faire nous allons préparer une nouvelle machine virtuelle sur laquelle nous allons déployer notre solution.

Une fois l'installation de notre nouvelle machine terminée, nous pouvons commencer.

```
Debian GNU/Linux 12 zabbix-server tty1
zabbix-server login: zabbix
Password:
Linux zabbix-server 6.1.0-20-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 4 20:40:55 EDT 2024 from 192.168.1.1 on pts/0
zabbix@zabbix-server:~$ _
```

## Préparation de l'environnement

Avant de commencer l'installation de Zabbix, on va avoir besoin d'un système de base de données ici nous utiliserons la base de données MariaDB.

Mais de commencer l'installation nous allons mettre à jour votre système d'exploitation et ensuite installer la base de données :

1. `sudo apt update`
2. `sudo apt upgrade`
3. `sudo apt install mariadb-server`

## Sécuriser l'installation MariaDB

1. `sudo mariadb-secure-installation`

Ensuite vous suivez les instructions et définissez un mot de passe (ce mot de passe nous l'utiliserons plus tard)

## Installation de Zabbix

Pour l'installation suivez les instructions du site officiels [ici](#)

Vous pouvez aussi suivre ce lien pour faire l'installation

<https://fr.linux-console.net/?p=1395>

Si à l'étape de l'installation vous avez cette erreur :

```
root@voipserver:/home/voip# dpkg -i zabbix-release_6.4-1+debian12_all.deb
dpkg: warning: 'ldconfig' not found in PATH or not executable
dpkg: warning: 'start-stop-daemon' not found in PATH or not executable
dpkg: error: 2 expected programs not found in PATH or not executable
Note: root's PATH should usually contain /usr/local/sbin, /usr/sbin and /sbin
```

Pour résoudre le problème :

1. créer un fichier avec touch et nomme le comme tu veux (`touch bashrc`)
2. ensuite edit le fichier avec nano (`nano bashrc`)
3. Copier et coller dans le fichier ceci :  
`export PATH=$PATH:/usr/local/sbin:/usr/sbin:/sbin`

```
GNU nano 7.2          bashrc *
```

```
export PATH=$PATH:/usr/local/sbin:/usr/sbin:/sbin
```

4. Ensuite enregistrer et fermer le fichier
5. changer le nom du fichier en "**~/.bashrc**"  
mv bashrc **~/.bashrc**
6. chmod +x **~/.bashrc**
7. source **~/.bashrc**

```
root@voipserver:/home/voip# mv bashrc ~/.bashrc
root@voipserver:/home/voip# chmod +x ~/.bashrc
root@voipserver:/home/voip# source ~/.bashrc
root@voipserver:/home/voip#
```

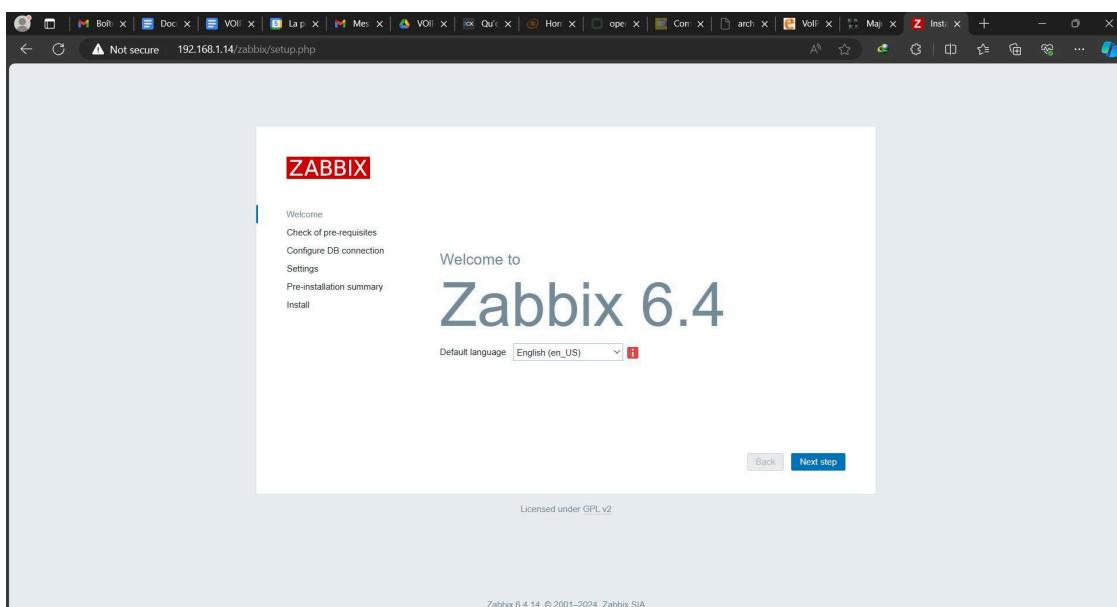
8. Maintenant relance l'installation et cette fois-ci tout devrait bien se passer

```
root@voipserver:/home/voip# dpkg -i zabbix-release_6.4-1+debian12_all.deb
Selecting previously unselected package zabbix-release.
(Reading database ... 52229 files and directories currently installed.)
Preparing to unpack zabbix-release_6.4-1+debian12_all.deb ...
Unpacking zabbix-release (1:6.4-1+debian12) ...
Setting up zabbix-release (1:6.4-1+debian12) ...
root@voipserver:/home/voip#
```

## Allons sur l'interface web

Une fois l'installation terminée, nous allons accéder à notre tableau de bord et le configurer pour notre première utilisation.

Pour y accéder : <http://192.168.1.14/zabbix> , remplacer (**192.168.1.14** par celui de votre serveur zabbix)





Check of pre-requisites				
PHP libxml	2.9.14	2.6.15	OK	
PHP xmlwriter	on		OK	
PHP xmlreader	on		OK	
PHP LDAP	on		OK	
PHP OpenSSL	on		OK	
PHP ctype	on		OK	
PHP session	on		OK	
PHP option "session.auto_start"	off	off	OK	
PHP gettext	on		OK	
PHP option "arg_separator.output"	&	&	OK	
System locale	en_US.utf8	en_US	OK	

At the bottom right are "Back" and "Next step" buttons.

Configure DB connection	
Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.	
Welcome	Database type <input type="button" value="MySQL"/>
Check of pre-requisites	Database host <input type="text" value="localhost"/>
Configure DB connection	Database port <input type="text" value="0"/> 0 - use default port
Settings	Database name <input type="text" value="zabbix"/>
Pre-installation summary	Store credentials in <input checked="" type="radio"/> Plain text <input type="radio"/> HashiCorp Vault <input type="radio"/> CyberArk Vault
Install	User <input type="text" value="zabbix"/>
	Password <input type="password" value="*****"/>
Database TLS encryption <small>Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).</small>	
<input type="button" value="Back"/> <input type="button" value="Next step"/>	

# ZABBIX

## Settings

Welcome  
Check of pre-requisites  
Configure DB connection  
Settings  
Pre-installation summary  
Install

Zabbix server name: Zabbix Server  
Default time zone: (UTC+02:00) Europe/Paris  
Default theme: Blue

Back Next step

# ZABBIX

## Pre-installation summary

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

Welcome  
Check of pre-requisites  
Configure DB connection  
Settings  
Pre-installation summary  
Install

Database type: MySQL  
Database server: localhost  
Database port: default  
Database name: zabbix  
Database user: zabbix  
Database password: \*\*\*\*\*  
Database TLS encryption: false

Zabbix server name: Zabbix Server

Back Next step

# ZABBIX

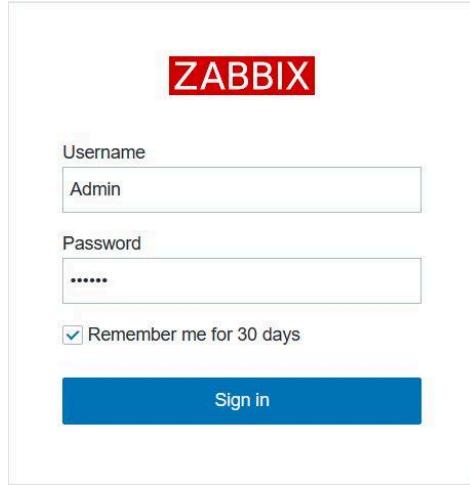
## Install

Welcome  
Check of pre-requisites  
Configure DB connection  
Settings  
Pre-installation summary  
Install

Congratulations! You have successfully installed Zabbix frontend.

Configuration file "conf/zabbix.conf.php" created.

Back Finish



Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	310	
Number of items (enabled/disabled/not supported)	110	99 / 0 / 11
Number of triggers (enabled/disabled [problem/ok])	64	64 / 0 [0 / 64]
Number of users (online)	2	1
Required server performance, new values per second	1.54	

## Maintenant retournons sur notre serveur Asterisk

Ici nous allons refaire les mêmes étapes mais sans installer le serveur complet, nous allons juste tout simplement installer **zabbix-agent** :

1. `wget`  
`https://repo.zabbix.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix-release_6.4-1+debian12_all.deb`
2. `dpkg -i zabbix-release_6.4-1+debian12_all.deb` ;(voir plus haut si erreur)
3. `apt update -y`
4. `apt install zabbix-agent -y`

```

root@voipserver:~#
root@voipserver:/home/voip# apt install zabbix-agent -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libmodbus5
The following NEW packages will be installed:
  libmodbus5 zabbix-agent
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 715 kB of archives.
After this operation, 1,219 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libmodbus5 amd64 3.1.6-2.1 [31.3 kB]
Get:2 https://repo.zabbix.com/zabbix/6.4/debian bookworm/main amd64 zabbix-agent amd64 1:6.4.14-1+debian12 [683 kB]
Fetched 715 kB in 2s (286 kB/s)
Selecting previously unselected package libmodbus5:amd64.
(Reading database ... 52235 files and directories currently installed.)
Preparing to unpack .../libmodbus5_3.1.6-2.1_amd64.deb ...
Unpacking libmodbus5:amd64 (3.1.6-2.1) ...
Selecting previously unselected package zabbix-agent.
Preparing to unpack .../zabbix-agent_1%3a6.4.14-1+debian12_amd64.deb ...
Unpacking zabbix-agent (1:6.4.14-1+debian12) ...
Setting up libmodbus5:amd64 (3.1.6-2.1) ...
Setting up zabbix-agent (1:6.4.14-1+debian12) ...
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service → /lib/systemd/system/zabbix-agent.service.
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+deb12u7) ...
root@voipserver:/home/voip#

```

## Editer le fichier de configuration Zabbix Agent avec nano

nano `/etc/zabbix/zabbix_agentd.conf`

Décommenter et remplacer les IP (`Server=127.0.0.1`, `ServerActive=127.0.0.1`) par celui du Serveur Zabbix et "Hostname=**Zabbix server**" par ce que vous voulez

```

# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=192.168.1.14

### Option: ListenPort
#       Agent will listen on this port for connections from the server.
#

```

```

# Mandatory: no
# Default:
# ServerActive=

ServerActive=192.168.1.14

### Option: Hostname
#       List of comma delimited unique, case sensitive hostnames.
#       Required for active checks and must match hostnames as configured on the server.
#       Value is acquired from HostnameItem if undefined.

```

```

# Mandatory: no
# Default:
# Hostname=

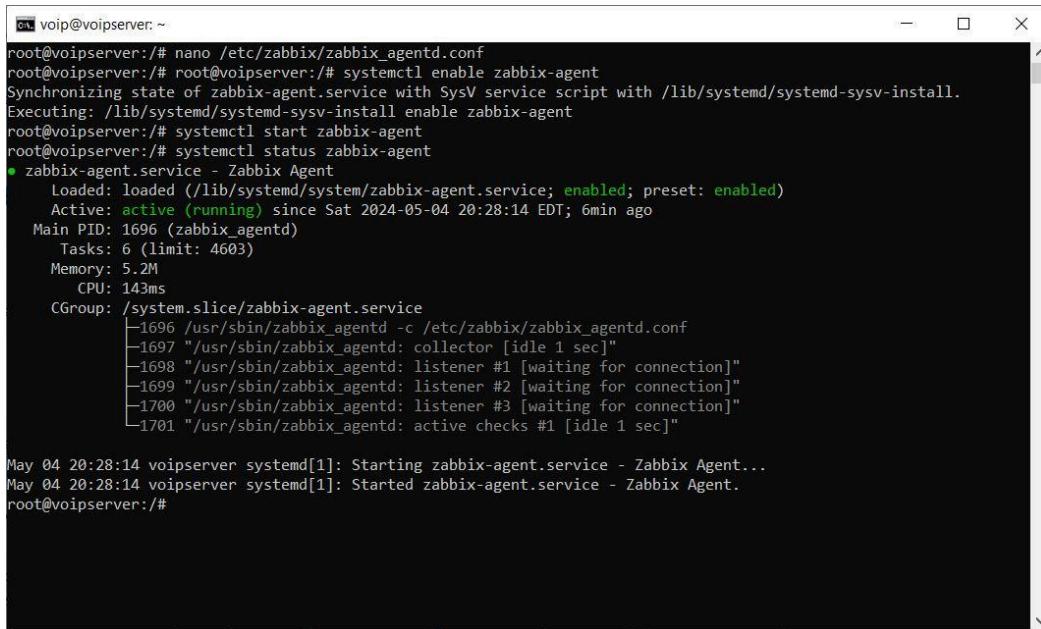
Hostname=Asterisk-SERVER

### Option: HostnameItem
#       Item used for generating Hostname if it is undefined. Ignored if Hostname is defined.
#       Does not support UserParameters or aliases.

```

## Activation et mise en marche de zabbix agent

1. `systemctl enable zabbix-agent`
2. `systemctl start zabbix-agent`
3. `systemctl status zabbix-agent`



```
root@voipserver:~# nano /etc/zabbix/zabbix_agentd.conf
root@voipserver:/# root@voipserver:/# systemctl enable zabbix-agent
Synchronizing state of zabbix-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
root@voipserver:/# systemctl start zabbix-agent
root@voipserver:/# systemctl status zabbix-agent
● zabbix-agent.service - Zabbix Agent
    Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; preset: enabled)
      Active: active (running) since Sat 2024-05-04 20:28:14 EDT; 6min ago
        Main PID: 1696 (zabbix_agentd)
          Tasks: 6 (limit: 4603)
        Memory: 5.2M
          CPU: 143ms
         CGroup: /system.slice/zabbix-agent.service
                 ├─1696 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
                 ├─1697 "/usr/sbin/zabbix_agentd: collector [idle 1 sec]"
                 ├─1698 "/usr/sbin/zabbix_agentd: listener #1 [waiting for connection]"
                 ├─1699 "/usr/sbin/zabbix_agentd: listener #2 [waiting for connection]"
                 ├─1700 "/usr/sbin/zabbix_agentd: listener #3 [waiting for connection]"
                 ├─1701 "/usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]"

May 04 20:28:14 voipserver systemd[1]: Starting zabbix-agent.service - Zabbix Agent...
May 04 20:28:14 voipserver systemd[1]: Started zabbix-agent.service - Zabbix Agent.
root@voipserver:/#
```

## Configuration Asterisk

`nano /etc/asterisk/manager.conf`

Modifiez les configurations suivantes :

```
[general]
enabled = yes
webenabled = yes

port = 5038
bindaddr = 0.0.0.0
```

et configurer les infos de l'utilisateur **AMI** (Asterisk Manager Interface) :

```
[zabbix]
secret = password1234
deny=0.0.0.0/0.0.0.0
permit=192.168.1.14/255.255.255.0
read = system,call,log,verbose,agent,user,config,dtmf,reporting,cdr,dialplan
write = system,call,agent,user,config,command,reporting,originate,message
```

1. `asterisk -rx "manager reload"`

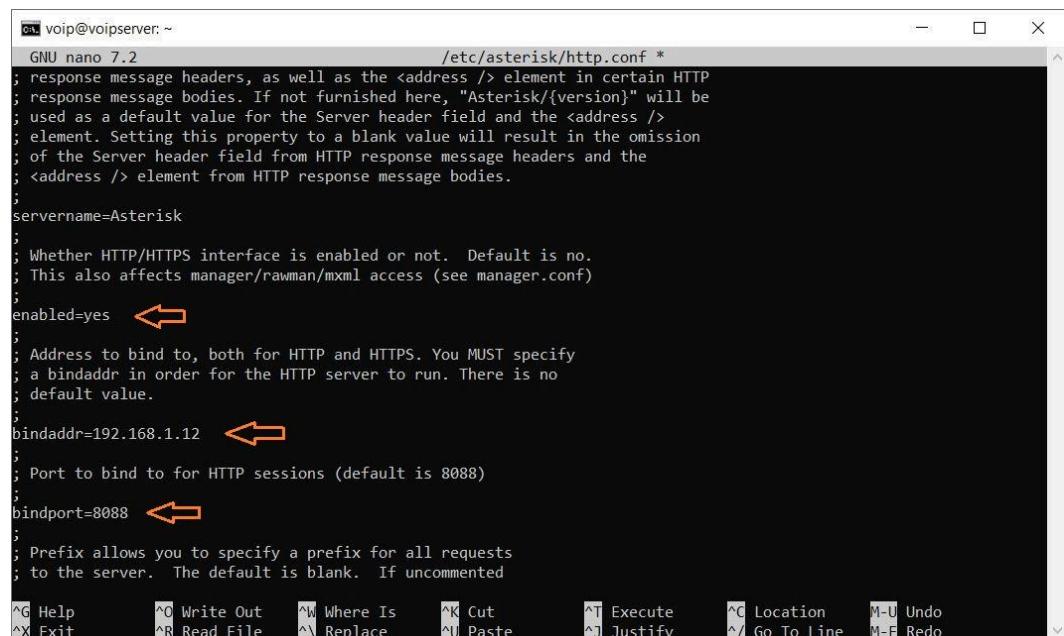
## 2. asterisk -rx "manager show users"

```
root@voipserver:/# nano /etc/asterisk/manager.conf
root@voipserver:/# asterisk -rx "manager reload"
root@voipserver:/# asterisk -rx "manager show users"

username
-----
zabbix
-----
1 manager users configured.
root@voipserver:/#
```

Modifiez les configurations suivantes : **bindaddr** = l'adresse IP de votre serveur Asterisk

## 1. nano /etc/asterisk/http.conf



```
GNU nano 7.2                               /etc/asterisk/http.conf *
; response message headers, as well as the <address /> element in certain HTTP
; response message bodies. If not furnished here, "Asterisk/{version}" will be
; used as a default value for the Server header field and the <address />
; element. Setting this property to a blank value will result in the omission
; of the Server header field from HTTP response message headers and the
; <address /> element from HTTP response message bodies.
;
servername=Asterisk
;
; Whether HTTP/HTTPS interface is enabled or not. Default is no.
; This also affects manager/rawman/mxml access (see manager.conf)
;
enabled=yes ←
;
; Address to bind to, both for HTTP and HTTPS. You MUST specify
; a bindaddr in order for the HTTP server to run. There is no
; default value.
;
bindaddr=192.168.1.12 ←
;
; Port to bind to for HTTP sessions (default is 8088)
;
bindport=8088 ←
;
; Prefix allows you to specify a prefix for all requests
; to the server. The default is blank. If uncommented
;
^G Help          ^O Write Out    ^W Where Is    ^K Cut          ^T Execute    ^C Location   M-U Undo
^X Exit          ^R Read File   ^L Replace     ^U Paste        ^J Justify    ^/ Go To Line M-E Redo
```

**asterisk -rx "reload"**

## Retour sur le serveur Zabbix

Autorisez le trafic provenant de votre serveur Zabbix :

1. sudo apt-get install iptables
2. iptables -I INPUT -s 192.168.1.12 -j ACCEPT

Remplacer l'IP par celui du serveur asterisk !!

The screenshot shows the Zabbix Global view dashboard. On the left, the navigation menu is open, with 'Data collection' selected. Under 'Hosts', there is a count of 2 hosts. The main area displays system information, including CPU utilization (1.53 Values per second), problems by severity (0 Disaster, 0 High, 0 Average, 0 Warning, 0 Information, 0 Not classified), and a geomap of Riga, Latvia. A large digital clock in the top right corner shows 02:45 Paris.

The screenshot shows the 'New host' dialog box. The 'Host' tab is selected. The 'Host name' field is highlighted with a red border. Other fields include 'Visible name', 'Templates', 'Host groups', and 'Interfaces'. Below these are 'Description' and 'Monitored by proxy' (set to '(no proxy)'). At the bottom right of the dialog, there are 'Add' and 'Cancel' buttons. The 'Create host' button in the top right of the main Zabbix window is also highlighted with a red border.

**Templates**

Template group

Name

Specify some filter condition to see the values.

**Template groups**

Name

Templates

Templates/Applications

Templates/Cloud

Templates/Databases

Templates/Network devices

Templates/Operating systems

Templates/Power

Templates/SAN

Templates/Server hardware

Templates/Telephony

Templates/Video surveillance

**Templates**

Template group

Name

AIX by Zabbix agent

FreeBSD by Zabbix agent

HP-UX by Zabbix agent

Linux by Prom

Linux by SNMP

Linux by Zabbix agent

Linux by Zabbix agent active

macOS by Zabbix agent

OpenBSD by Zabbix agent

Solaris by Zabbix agent

Windows by SNMP

Windows by Zabbix agent

Windows by Zabbix agent active

**Templates**

Template group

Name  
 Asterisk by HTTP

**Host groups**

Name  
 Applications  
 Databases  
 Discovered hosts  
 Hypervisors  
 Linux servers  
 Virtual machines  
 Zabbix servers

**New host**

Host IPMI Tags Macros Inventory Encryption Value mapping

\* Host name   
Visible name   
Templates    
type here to search

\* Host groups    
type here to search

Interfaces No interfaces are defined.

Add

Description	Agent
	SNMP
	JMX
	IPMI

Monitored by proxy

Enabled

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent	192.168.1.12			IP	DNS	10050

Remove

New host

Host IPMI Tags Macros Inventory Encryption Value mapping

Host macros Inherited and host macros

Macro	Value	Description
[\$MACRO]	value	T description Remove

Add

Add Cancel

Host macros Inherited and host macros

[\$AMI.SECRET]	password1234	T Remove ← Asterisk by HTTP: "zabbix"
[\$AMI.URL]	http://192.168.1.12:8088/asterisk/rawman	T Remove ← Asterisk by HTTP: "http://asterisk:8088/asterisk/rawm..."

Not secure | 192.168.1.14/zabbix/zabbix.php?action=host.list

Hosts

Host added

Host groups: type here to search Select Status Any Enabled Disabled

Templates: type here to search Select Monitored by Any Server Proxy

Name: DNS: IP: Port: Proxy: Tags: And/Or Or Add

Apply Reset

Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
Asterisk.SERVER	Items 69	Triggers 24	Graphs 12	Discovery 7	Web	192.168.1.12:10050		Asterisk by HTTP, Linux by Zabbix agent	Enabled	ZBX	None		
Zabbix server	Items 128	Triggers 69	Graphs 24	Discovery 5	Web	127.0.0.1:10050		Linux by Zabbix agent, Zabbix server health	Enabled	ZBX	None		

Displaying 2 of 2 found

0 selected Enable Disable Export Mass update Delete

Zabbix 6.4.14, © 2001–2024, Zabbix SIA

Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
Asterisk.SERVER	Items 75	Triggers 29	Graphs 13	Discovery 7	Web	192.168.1.12:10050		Asterisk by HTTP, Linux by Zabbix agent	Enabled	ZBX	None		
Zabbix server	Items 128	Triggers 69	Graphs 24	Discovery 5	Web	127.0.0.1:10050		Linux by Zabbix agent, Zabbix server health	Enabled	ZBX	None		

0 selected Enable Disable Export Mass update Delete

**Global view**

Dashboard updated

**Top hosts by CPU utilization**

	Utilization	1m avg	5m avg	15m avg	Processes
Zabbix server	0.73 %	0.01	0.02	0.00	199
Asterisk SERVER	0.08 %	0.07	0.12	0.10	139

**System information**

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	2 / 2	0 / 0
Number of templates	310	
Number of items (enabled/disabled/not supported)	203 / 0 / 35	
Number of triggers (enabled/disabled [problem/ok])	98 / 98 [0 / 98]	
Number of users (online)	2 / 1	

**Host availability**

Available	2
Not available	0
Unknown	0
Total	2

**Problems by severity**

Disaster	0
High	0
Average	0
Warning	0
Information	0
Not classified	0

**Current problems**

No data found.

**Geomap**

**System performance**

All hosts / Asterisk-SERVER / System performance

From: now-1h To: now

Last 2 days Yesterday Today Last 5 minutes  
Last 7 days Day before yesterday Today so far Last 15 minutes  
Last 30 days This day last week This week Last 30 minutes  
Last 3 months Previous week This week so far Last 1 hour  
Last 6 months Previous month This month Last 3 hours  
Last 1 year Previous year This month so far Last 6 hours  
Last 2 years Last 2 years This year Last 12 hours  
This year so far Last 1 day

**Linux: System load**

Legend:

- Linux: Load average (1m avg) [avg]: 0.04492
- Linux: Load average (5m avg) [avg]: 0.09277
- Linux: Load average (15m avg) [avg]: 0.08691
- Linux: Number of CPUs [avg]: 4

Table:

	last	min	avg	max
Linux: Load average (1m avg) [avg]	0.04492	0.02441	0.00381	0.1997
Linux: Load average (5m avg) [avg]	0.09277	0.09277	0.1209	0.1548
Linux: Load average (15m avg) [avg]	0.08691	0.08691	0.09678	0.1089
Linux: Number of CPUs [avg]	4	4	4	4

**Linux: CPU usage**

Legend:

- Linux: CPU user nice time [avg]: 0 %
- Linux: CPU guest nice time [avg]: 0 %
- Linux: CPU guest time [avg]: 0 %
- Linux: CPU softirq time [avg]: 0.03341 %
- Linux: CPU interrupt time [avg]: 0 %
- Linux: CPU steal time [avg]: 0 %
- Linux: CPU system time [avg]: 0.00417 %
- Linux: CPU nice time [avg]: 0 %
- Linux: CPU user time [avg]: 0.008352 %
- Linux: CPU system time [avg]: 0.02088 %

Table:

	last	min	avg	max
Linux: CPU guest nice time [avg]	0 %	0 %	0 %	0 %
Linux: CPU guest time [avg]	0 %	0 %	0 %	0 %
Linux: CPU softirq time [avg]	0.03341 %	0.02508 %	0.03822 %	0.05008 %
Linux: CPU interrupt time [avg]	0 %	0 %	0 %	0 %
Linux: CPU steal time [avg]	0 %	0 %	0 %	0 %
Linux: CPU system time [avg]	0.00417 %	0 %	0.00372 %	0.00611 %
Linux: CPU nice time [avg]	0 %	0 %	0 %	0 %
Linux: CPU user time [avg]	0.008352 %	0.004173 %	0.006354 %	0.01089 %
Linux: CPU system time [avg]	0.02088 %	0.02087 %	0.02645 %	0.03344 %

**Linux: Memory usage**

Save GB

1536 MB