

VoIP

ĆWICZENIE Asterisk

Dariusz Dwornikowski
20 maja 2010

1. Wprowadzenie

Ćwiczenia mają na celu wprowadzenie do konfiguracji podstawowych elementów budujących infrastrukturę SIP, tj.:

- serwer SIP Proxy,
- serwer rejestracji SIP,
- serwer dostępności (ang. *SIP presence server*),
- klient (ang. *user agent*).

Serwer Asterisk jest często nazywany kombajnem telekomunikacyjnym. Oznacza to, że potrafi on obsługiwać wiele protokołów komunikacyjnych, takich jak: SIP, ISDN, IAX, H.323, DSS1. Wpiera także obsługę kart sprzętowych, umożliwiających podłączenie różnych mediów (telefonów, linii ISDN, linii PSTN, itd). W przypadku protokołu SIP, Asterisk potrafi pełnić rolę każdego elementu infrastruktury SIP.

1.1. Podstawowe informacje

Wszystkie pliki konfiguracyjne serwera Asterisk znajdują się w katalogu: `/etc/asterisk`. Włączenie serwera odbywa się poprzez wywołanie komendy `asterisk`. Część statycznej konfiguracji Asteriska odbywa się przez edycję plików konfiguracyjnych, jednakże możliwe jest także zarządzanie serwerem w czasie rzeczywistym poprzez interaktywne CLI, uruchamiane przez: `asterisk -r`.

Komendy CLI:

`core set verbose 3` ustawienie poziomu interaktywności wiadomości diagnostycznych Asteriska

`reload` przeładowanie konfiguracji z plików konfiguracyjnych

`core restart now` przeładowanie procesu serwera

`sip set debug on` włączenie trybu debugu protokołu SIP

`sip show peers` wylistowanie użytkowników zarejestrowanych w Asterisku

`sip show registry` wylistowanie kont, na które Asterisk jest zarejestrowany

`sip show channels` wylistowanie aktywnych kanałów SIP

`show dialplan` wylistowanie aktualnego dialplanu

1.2. Konfiguracja użytkowników

Użytkowników SIP konfiguruje się w pliku konfiguracyjnym Asteriska `sip.conf`. Aby stworzyć użytkownika należy stworzyć dla niego wpis, ustawić jego nazwę, ewentualnie hasło oraz odpowiednie dyrektywy, dotyczące kodeków, prezentacji oraz monitorowania. Poniższy przykład prezentuje przykładowy wpis dla użytkownika:

```
[user]
type=friend
username=user
secret=password
insecure=very
; Dla pewności dajemy nat
```

```
nat=yes
host=dynamic ; może się rejestrować z różnych adresów
qualify=20
; Ustawienia dot. połączeń
canreinvite=no
dtmfmode=info
; Ustawienia kodeków (pozwalamy na wszystkie)
allow=all
; Ustawienie kontekstu w dialplanie (extensions.conf)
context=default
```

1.3. Konfiguracja “trunk”

W pliku konfiguracyjnym `sip.conf` znajdują się ustawienia Asteriska dotyczące obsługi protokołu SIP. Aby zarejestrować w Asterisku użytkownika zewnętrznego (np. konto uzyskane od dostawcy usług VoIP), należy skorzystać z dyrektywy `register`. Ustawienie to pozwoli na odbieranie połączeń przychodzących do Asteriska na “numerze” `extension`. Poniższy listing przedstawia przykładową konfigurację.

```
register => user:password@sip.example.com/extension
```

Po wpisaniu dyrektywy `register`, należy przeładować serwer Asterisk, używając komendy `reload`. Swoje ustawienia można zweryfikować wykonując komendę: `sip show registry`.

Aby móc wykonywać połączenia wychodzące przez konto `user`, konieczne jest dodanie użytkownika, który będzie zarejestrowany w serwerze Asterisk. Dane uwierzytelniające są takie same jak w przypadku listingu powyżej. Powiązanie dyrektyw `register` oraz stworzenie użytkownika pozwala na osiągnięcie pełnej funkcjonalności trunka (kanału zewnętrznego, linii zewnętrznej). Poniższy listing prezentuje przykładową konfigurację:

```
[user]
type=peer
username=user
fromdomain=sip.example.com
fromuser=user
secret=passsword
insecure=very
; Dla pewności dajemy nat
nat=yes
host=dynamic
qualify=20
; Ustawienia dot. połączeń
canreinvite=no
dtmfmode=info
; Ustawienia kodeków (pozwalamy na wszystkie)
allow=all
; Ustawienie kontekstu w dialplanie (extensions.conf)
context=default
```

Analogicznie jak dla wcześniejszego przykładu, Asterisk musi zostać przeładowany. Zweryfikowanie ustawień można zweryfikować komendą: `sip show peers`.

1.4. Konfiguracja “dialplan”

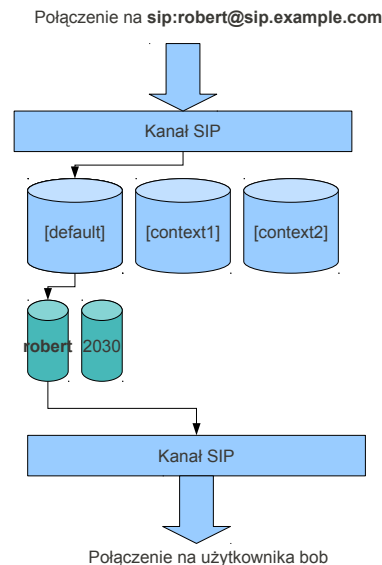
Dialplan jest rdzeniem Asteriska odpowiedzialnym za przetwarzanie połączeń przechodzących przez serwer. Można o nim myśleć jak o elemencie trasującym. Definicja dialplanu znajduje się w pliku konfiguracyjnym `extensions.conf`. Składa się on z kontekstów (ang. `context`), definiowanych w kwadratowych nawiasach, np. `[default]`. Konteksty są blokami dialplanu, do których kierowane są połączenia. W kontekstach definiuje się końcówki (`extensions`), można o nich myśleć

jak o numerach wewnętrznych w centrali abonenckiej. Każda końcówka składa się z identyfikatora, priorytetu oraz komendy Asteriska. Priorytet definiuje kolejność przetwarzania komend. Poniższy przykład ukazuje kontekst domyślny ze zdefiniowanymi dwoma końcówkami - **robert** oraz **2030**.

```
[default]
; składania końcówki
; exten => nazwa, priorytet, komenda
exten => robert, 1, Answer
exten => robert, 2, Playback(beep)
exten => robert, 3, Dial(SIP/bob,20,rRtg)
exten => robert, 4, Hangup

exten => 2030, 1, Answer
exten => 2030, n, Dial(SIP/bob,20,rRtg)
exten => 2030, n, Hangup
```

W przykładzie zarówno połączenia na adres **sip:robert@sip.example.com**, jak i **sip:2030@sip.example.com** zostaną przekierowane na klienta SIP **bob**, jeśli jest zarejestrowany w serwerze Asterisk. Komenda **Dial** wykonuje połączenie w technologii SIP na konto **bob**, **timeout** jest ustawiony na 20s, dodatkowe parametry to **rRtg**. Rys. 1 obrazuje ogólną zasadę działania dialplanu.



Rys. 1: Dialplan w Asterisk

Z kanałem (połączeniem) w Asterisku skojarzony jest szereg zmiennych kanału. Przykładowymi zmiennymi są:

CALLERID(name) nazwa dzwoniącego, np. "Jan Kowalski"

CALLERID(num) numer dzwoniącego lub konto, np. "jan_kowalski"

CONTEXT aktualny kontekst

SIPDOMAIN domena SIP

CHANNEL nazwa kanału

ANSWEREDTIME aktualny czas trwania rozmowy

EXTEN aktualna końcówka – “to na co dzwोनiono”

Istnieją także zmienne skojarzone z aplikacjami, np. aplikacja **Dial** posiada skojarzoną ze sobą zmienną **DIALSTATUS**, która przyjmuje wartość rezultatu połączenia. Jest to wygodny sposób, aby sprawdzić, czy rozmowa przebiegła pomyślnie, czy też wystąpił nieoczekiwany błąd. Aby móc odczytać wartość tej zmiennej, konieczne jest podanie argumentu **g** do komendy **Dial**, tak aby przetwarzanie po zakończeniu rozmowy mogło kontynuować w ramach dialplanu.

```
[default]
exten => 2020, 1, Answer()
exten => 2020, 2, Dial(SIP/zdzisława,20,g)
exten => 2020, 3, Noop(${DIALSTATUS})
exten => 2020, 4, Hangup()
```

Inną przydatną zmienną jest **EXTEN**, która przybiera zawsze wartość końcówki, na którą trafiła rozmowa. Pozwala to na dynamiczne odczytanie np. numery, na który dzwонiono i jego zmianę, czy weryfikację przed dokonaniem przekierowania. Poniższy przykład dokonuje wyświetlenia numeru (nazwy), na który przychodzi połączenie. Zmienna **_X.** to wzorzec, który określa, że końcówka może przybierać dowolne wartości numeryczne.

```
[default]
exten => _X., 1, Answer()
exten => _X., n, Noop(Dzwонiono na ${EXTEN})
exten => _X., n, Hangup
```

Linki:

- Komendy i aplikacje Asterisk — <http://www.voip-info.org/wiki/view/Asterisk+-+documentation+of+application+commands>
- Dialplan — <http://www.voip-info.org/wiki/index.php?page=Asterisk+config+extensions.conf>
- Zmienne kanału — <http://www.voip-info.org/wiki/view/Asterisk+variables>
- Wzorce końcówek — <http://www.voip-info.org/wiki/view/Asterisk+Dialplan+Patterns>

1.5. AGI (Asterisk Gateway Interface)

Asterisk Gateway Interface (AGI) jest interfejsem programistycznym dla serwera Asterisk, pozwala na podłączenie się dowolnym programem implementującym AGI pod przebieg rozmowy i dowolnie ją sterowanie. Z poziomu AGI dostępne są wszystkie komendy i aplikacje z dialplanu. Przykładowo, można odebrać rozmowę, odegrać zapowiedź głosową i zadzwонić na inny kanał (np. wykonanie aplikacji **Dial**).

Skrypty AGI umieszczone są w katalogu `/var/lib/asterisk/agi-bin/`, mogą posiadać dowolne rozszerzenie. Ważnym jest aby pamiętać, że skrypty te muszą mieć prawa do wykonania dla użytkownika, na którym działa serwer Asterisk (przeważnie **root**).

Metoda połączenia do AGI dzieli się na dwa rodzaje: lokalną oraz zdalną. W wersji pierwszej skrypty AGI uruchamiane są przez serwer Asterisk na tym samym węźle, jako zewnętrzne programy. Większość prostych scenariuszy zakłada właśnie takie użycie protokołu. Z uwagi na problemy wydajnościowe stworzono wersję zdalną protokołu, tzw. FastAGI, komunikujące się z serwerem Asterisk za pomocą protokołu IP. Wiadomości AGI przesyłane są do zdalnego hosta, gdzie odbywa się przetwarzanie.

Programy AGI wywoływane są z poziomu dialplanu komendą **AGI**, lub **DeadAGI**, różnica polega na tym, iż skrypty **DeadAGI** nie kończą przetwarzania po zamknięciu kanału. Jest to wykorzystywane np. w sytuacjach, gdy rekordy billingowe zrucane są po zakończeniu rozmowy, lub zapisywane są dane rozmowy (np. prepaid). Poniższy przykład prezentuje wywołanie programu AGI, zarówno lokalnego, jak i zdalnego:

```
exten => 2030,1,AGI(plik.agi)
exten => 2030,1,AGI(agi://192.168.1.1)
```

Dużą zaletą AGI jest istnienie bibliotek programistycznych niemalże dla każdego znanego języka programowania, np. C, C#, Java, Perl, Python, PHP, Lisp itd. Najczęściej spotykane skrypty lokalne pisane są w języku Perl oraz PHP, ze względu na ich upowszechnienie. Dla AGI wykonywanego zdalnie przeważnie stosowane są własne rozwiązania lub serwery w języku Java oraz C#. Poniższe przykłady demonstrują proste programy w języku Perl oraz PHP.

```
#!/usr/bin/perl
use Asterisk::AGI;
my $agi = new Asterisk::AGI();
my %input = $agi->ReadParse();

$agi->answer();
$agi->stream_file("hello",'1');
$agi->hangup();
```

```
<?php
include("phpagi.php");
$agi = new AGI();
$agi->answer();
$agi->stream_file("hello");
$agi->hangup();
?>
```

Linki

1. PHPAGI — <http://phpagi.sourceforge.net/>
2. Perl AGI — <http://www.voip-info.org/tiki-index.php?page=Asterisk+perl+agi>
3. AGI — <http://www.voip-info.org/wiki/view/Asterisk+AGI>
4. Perl AGI — <http://asterisk.gnuinter.net/files/asterisk-perl-1.01.tar.gz>

2. Zadania

1. Skonfiguruj serwer Asterisk, aby obsługiwał protokół SIP, oraz dodaj użytkownika. Sprawdź czy użytkownik został poprawnie dodany.
2. Zarejestruj użytkownika przy pomocy programu Linphone oraz bramki PAP2 lub telefonu IP.
3. Zarejestruj kilku użytkowników oraz wykonaj połączenia wewnętrzne pomiędzy nimi. Uruchoom tryb debugu w serwerze Asterisk (`core set verbose 10`) oraz popatrz co się dzieje w czasie wykonywania rozmów.
4. Stwórz kontekst w serwerze Asterisk oraz spraw by rozmowy przychodzące do użytkownika zdefiniowanego w `sip.conf` trafiały do tegoż kontekstu. Wykonaj operację odegrania pliku dźwiękowego znajdującego się w `/var/lib/asterisk/sounds`.
5. Wykonaj przekierowanie rozmowy: gdy zajęty, gdy nie odbiera, gdy niedostępny oraz bezwarunkowe.
6. Korzystając z komendy NoOp, wyświetl dane kanału oraz nagłówki SIP.
7. Skonfiguruj dwa serwery Asterisk, niech pierwszy z nich będzie trunkiem dla drugiego. Wykonaj połączenia przez trunk, oraz odbierz połączenia przychodzące.
8. Połącz trzy serwery Asterisk i zasymuluj LCR.



9. Napisz prosty program AGI, symulujący menu jednostopniowe.
10. Napisz program pobierający numer telefonu i wykonujący na niego przekierowanie.
11. Stwórz skrypt, który oddzwania na numer, z którego ostatnio dzwoniono, po czym odgrywa zapowiedź głosową z datą.
12. Stwórz program konferencji z pinem wykorzystaniem komendy `Meetme` oraz AGI.