

Korištenje Rsync uslužnog programa za sinhronizaciju podataka između servera

Studentski rad

Marko Marinković

student drugog ciklusa studija

Univerzitet u Istočnom Sarajevu, Elektrotehnički fakultet

Istočno Sarajevo, Bosna i Hercegovina

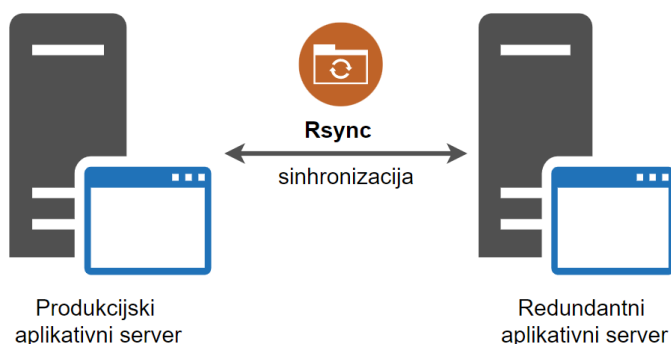
markos_m85@yahoo.com

Sažetak — U ovom radu biće objašnjeno korištenje Rsync uslužnog programa za replikaciju datoteka i podataka između dva aplikativna servera. U pitanju su serveri koji se nalaze na različitim lokacijama na kojima je instaliran Oracle Fusion Middleware. Na primarnoj lokaciji se nalazi produkcijski server dok se na sekundarnoj lokaciji nalazi njegova kopija čiji je zadatak da obezbijedi oporavak sistema od katastrofe (engl. *DR-Disaster Recovery*). Pored Rsync uslužnog programa korišteni su Cron servis i SSH protokol (engl. *Secure Shell Protocol*), čime se postiže automatizacija procesa sinhronizacije podataka između dva servera.

Ključne riječi - Rsync uslužni program; Cron servis; SSH protokol

I. UVOD

Kada imamo sistem koga čine primarni produkcijski server i redundantni server koji je njegova kopija smještena na različitoj geografskoj lokaciji, potrebno je obezbijediti da su u svakom trenutku podaci sinhronizovani. Arhitektura sistema koja je prikazana na Sl. 1 sastoji se od primarnog produkcijskog aplikativnog servera na kojem je instaliran Oracle-ov softver Oracle Forms and Reports koji služi za izradu izvještaja koristeći podatke pohranjene u bazi na udaljenom serveru.



Slika 1. Arhitektura sistema

Na drugoj geografskoj lokaciji se nalazi aplikativni server sa identičnom instalacijom Oracle Forms and Reports softvera čiji zadatak je da u slučaju katastrofe (požar, fizički kvar na serveru, vremenske nepogode i sl.) preuzme funkciju primarnog produkcijskog servera na taj način da krajnji korisnik može nesmetano nastaviti sa radom kao da do katastrofe nije ni došlo.

Na primarnoj lokaciji na produkcijskom serveru postoji folder sa datotekama koje koristi Oracle Forms and Reports i podložne su stalnim promjenama. Identičan folder postoji i na sekundarnoj strani i potrebno je obezbijediti, što je približnije moguće, da podaci u oba foldera budu identični.

Sinhronizacija podataka se postiže korištenjem Rsync uslužnog programa. Pored ovog uslužnog programa korišten je i Cron servis koji je zadužen za pokretanje Rsync programa u unaprijed definisano vrijeme. Prenos podataka između dva servera odvija se korištenjem SSH protokola što znači da se u toku prenosa podataka podaci šifriraju. Osim šifriranja podataka u toku prenosa podataka, SSH protokol omogućava pristupanje sa produkcijskog servera na redundantni server bez upotrebe lozinki (engl. *credentials*).

II. RSYNC USLUŽNI PROGRAM, CRON SERVIS I SSH PROTOKOL

A. Rsync uslužni program

Rsync je brz i izuzetno svestran alat za kopiranje datoteka. Može kopirati lokalno, na host ili sa drugog hosta koristeći SSH protokol. Nudi veliki broj opcija koje kontrolišu svaki aspekt njegovog ponašanja i dozvoljavaju vrlo fleksibilnu specifikaciju skupa datoteka koje treba kopirati.

Poznat je po svom delta-transfer algoritmu, koji smanjuje količinu podataka koji se šalju preko mreže šaljući samo razlike između izvornih datoteka i postojećih datoteka na odredištu. Rsync se koristi za izradu rezervnih kopija i preslikavanje (engl. *mirroring*) te kao poboljšana naredba za kopiranje [1].

Ovaj uslužni program je puno više od obične naredbe za kopiranje i ovo su neke od mogućnosti koje pruža:

- može ispravno kopirati sve podatke od izvora do odredišta, uključujući posebne datoteke i sve odgovarajuće dozvole (engl. *permission*).
- Podrazumijevani mehanizam provjere autentičnosti je SSH protokol, ali se to može lako zaobići promjenom varijable `RSYNC_RSH` u `rsh`.
- Osim provjere autentičnosti putem `RSH` i `SSH`, `Rsync` se takođe može pokrenuti kao pozadinski servis (engl. *daemon*) bilo u autentifikovanom ili anonimnom načinu rada. Prvi pruža sigurniji mehanizam za autentifikaciju, a drugi je veoma pogodan za preslikavanje.
- Može isključiti datoteke na isti način kako to čini GNU `tar`, koristeći stringove za isključenje na komandnoj liniji ili kreiranjem datoteke za izuzimanje ili pomoću opcija isključenja [2].
- Prilikom ažuriranja odredišta, izvor i odredište dijele svaku promijenjenu datoteku u blokove i pokreću dvije CRC provjere (engl. *Cyclic Redundancy Check*) provjeravajući svaki blok. Prenose se samo oni blokovi podataka koji se nakon CRC provjere ne slažu.
- Može izbrisati datoteke na odredištu koje više nisu prisutne na izvoru.

B. Cron servis

`Cron` je uslužni program za operativne sisteme bazirane na Unix-u koji omogućava da se zadaci automatski pokreću u pozadini u redovnim intervalima od strane pozadinskog servisa. `Crontab` (engl. *CRON TABLE*) je datoteka koja sadrži raspored `Cron` unosa koji će se pokrenuti u određeno vrijeme [3].

Lokacija datoteke varira u zavisnosti od operativnog sistema. Sadržaj `crontab` datoteke predstavljen je u sljedećem formatu:

```
*****zadatak (1)
```

gdje svaka zvjezdica (*) predstavlja vremenske periode u kojima se izvode naredbe. Svi periodi i njihove moguće vrijednosti iz naredbe (1) su navedeni u Tabeli I.

TABELA I. VREMENSKI PERIODI U CRONTAB DATOTECI

Kolona	Vrijednost	Opis
Minute	0-59	Minuta u kojoj će se izvršiti određena naredba
Sati	0-23	Sat u kojem će se izvršiti određena naredba
Dani	1-31	Dan u mjesecu u kojem će se izvršiti određena naredba
Mjeseci	1-12	Mjesec u godini u kojem će se izvršiti određena naredba
Dani u sedmici	0-6	Dan u sedmici u kojem će se izvršiti određena naredba

U Tabeli I. su navedeni vremenski periodi koji se koriste za definisanje `Cron` poslova (engl. *job*) koji će se izvršavati u određeno vrijeme. `Crontab` naredba uz navođenje odgovarajućih parametara se koristi za izmjenu `crontab` datoteke [4].

C. SSH protokol

Protokol radi u klijent-server režimu rada, što znači da vezu uspostavlja SSH klijent koji se povezuje na SSH server. SSH klijent pokreće proces uspostavljanja veze i koristi kriptografiju javnog ključa za provjeru identiteta SSH servera. Nakon faze podešavanja, SSH protokol koristi jaku simetričnu enkripciju i algoritme heširanja kako bi osigurao privatnost i integritet podataka koji se razmjenjuju između klijenta i servera [5].

SSH protokol se koristi za:

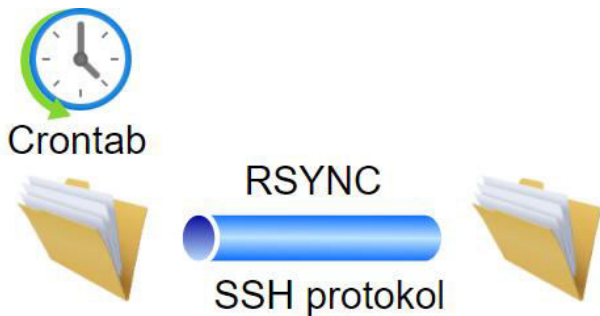
- obezbjeđivanje sigurnog pristupa za korisnike i automatizovane procese
- interaktivni i automatizovani prenos datoteka
- daljinsko pokretanje naredbi
- upravljanje mrežnom infrastrukturom i drugim kritičnim komponentama sistema.

Postoji nekoliko opcija koje se mogu koristiti za autentifikaciju korisnika. Najčešći su lozinke i provjera autentičnosti javnog ključa.

Metoda provjere autentičnosti javnog ključa prvenstveno se koristi za automatizaciju. Ideja je da postoji par kriptografskih ključeva (javni ključ i privatni ključ) i potrebno je konfigurisati javni ključ na serveru da autorizuje i odobri pristup serveru svakome ko ima kopiju privatnog ključa. Ključevi koji se koriste za autentifikaciju nazivaju se SSH ključevi [6].

III. SINHRONIZACIJA PODATAKA IZMEĐU SERVERA

U nastavku rada biće detaljno predstavljeno na koji način je realizovana sinhronizacija između dva foldera, što je prikazano na Sl. 2.



Slika 2. Sinhronizacija podataka korištenjem Cron servisa i SSH protokola

Potrebno je obezbijediti da Rsync uslužni program može sa primarnog servera putem SSH protokola pristupiti redundantnom serveru bez upotrebe lozinki, jer se cijeli proces odvija automatski bez uticaja korisnika. Ovo se postiže generisanjem para SSH ključeva.

A. Kreiranje SSH ključeva

Prenos podataka između servera će se vršiti upotrebom SSH protokola zbog čega je potrebno generisati SSH ključeve na primarnom serveru (*prod-issa*). Ovim se postiže da Rsync može pristupiti redundantnom serveru (*prod-dr-issa*) bez korištenja lozinke za autentifikaciju. Na primarnom serveru ćemo kreirati privatni i javni ključ koristeći *ssh-keygen* alat za kreiranje para ključeva, kao što je prizakano na Sl. 3.

```
192.168.56.173 - PuTTY
oracle@-prod-issa/home/oracle> ssh-keygen -t rsa
Generating public/private rsa key pair
Enter file in which to save the key (/home/oracle/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/oracle/.ssh/id_rsa.
Your public key has been saved in /home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:/9qbDsb0kd76y3fXPCVzXnJzK6FeJcVB5GarrLyy8eY oracle@-prod-issa
The key's randmart image is:
+----[RSA 2048]-----+
+----[SHA256]-----+
oracle@-prod-issa/home/oracle>
oracle@-prod-issa/home/oracle> cd .ssh
oracle@-prod-issa/home/oracle/.ssh> ls
id_rsa id_rsa.pub known_hosts
oracle@-prod-issa/home/oracle/.ssh>
```

Slika 3. Generisanje SSH ključeva

Kreirani su SSH ključevi i potrebno je javni ključ *id_rsa.pub* sa lokacije */home/oracle/.ssh* na primarnom serveru prekopirati na redundantni server na lokaciju

/home/oracle/.ssh kako bi se sa primarnog servera moglo pristupiti redundantnom serveru .

Nakon kreiranja SSH ključeva provjeravamo da li je moguće pomoću SSH protokola povezati se sa primarnog na redundantni server, kao što je prikazano na Sl. 4.

```
192.168.56.173 - PuTTY
oracle@-prod-issa/home/oracle> hostname
oracle@-prod-issa
oracle@-prod-issa/home/oracle> ssh 172.16.42.173
Last unsuccessful login: Tue Feb 16 12:07:05 2021 on ssh from 192.168.3.249
Last login: Mon Jan 17 00:08:12 2022 on ssh from 172.16.121.173
*****
*
* Welcome to AIX Version 7.2!
*
* Please see the README file in /usr/lpp/bos for information pertinent to
* this release of the AIX Operating System.
*
*****
oracle@-prod-dr-issa/home/oracle> hostname
oracle@-prod-dr-issa
```

Slika 4. Povezivanje produkcijskog servera sa redundantnim serverom korištenjem SSH protokola

Na Sl. 4 se vidi da smo se prvo logovali na produkcijski server i da smo se naredbom:

```
ssh 172.16.42.173 (2)
```

povezali sa redundantnim serverom, a da nam pri tome nije bila potrebna lozinka. Ovim je ispunjen uslov da pređemo na sljedeću fazu implementacije sinhronizacije podataka između servera, a to je konfigurisanje Rsync uslužnog programa.

B. Konfigurisanje Rsync uslužnog programa

Za početak potrebno je instalirati Rsync paket na produkcijskom i redundantnom serveru. Na Sl. 5 se vidi da su paketi instalirani na oba servera.

```
192.168.56.173 - PuTTY
oracle@-prod-issa/home/oracle> rpm -qa | grep rsync
rsync-3.2.3-1.ppc
oracle@-prod-issa/home/oracle>

172.16.42.173 - PuTTY
oracle@-prod-dr-issa/home/oracle> rpm -qa | grep rsync
rsync-3.2.3-1.ppc
oracle@-prod-dr-issa/home/oracle>
```

Slika 5. Instalirani Rsync paketi na produkcijskom i redundantnom serveru

Na produkcijskom serveru na lokaciji */u01/izvjestaji_prod* nalazi se folder koji sadrži .rdf datoteke koje će Oracle Forms and Reports koristiti za izradu izvještaja. Potrebno je obezbijediti da se svaka promjena u ovom folderu (kreiranje, ažuriranje ili brisanje datoteka) automatski prenese i u *izvjestaji_dr* folder na redundantnom serveru. To se postiže korištenjem Rsync uslužnog programa koji, kada se pokrene, provjerava stanje u folderu na primarnom serveru i na osnovu

njega ažurira stanje na određišanom serveru. Folderima *izvjestaji_prod* na produkcijskom i *izvjestaji_dr* na redundantnom serveru potrebno je dodijeliti dozvole kako bi im *Rsync* mogao pristupati. To se postiže sljedećim naredbama:

```
chmod 750 /u01/izvjestaji_prod (3)
```

```
chmod 750 /u01/izvjestaji_dr (4)
```

Sinhronizacija podataka između foldera se vrši pokretanjem sljedeće naredbe:

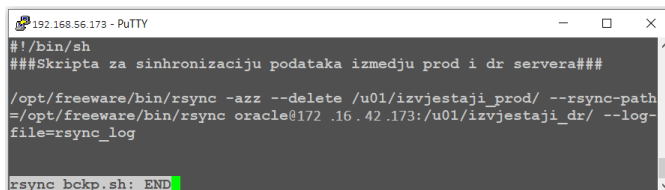
```
/opt/freeware/bin/rsync -azz --delete /u01/izvjestaji_prod/
--rsync-path=/opt/freeware/bin/rsync (5)
oracle@172.16.42.173:/u01/izvjestaji_dr/
--log-file=rsync_log
```

Kada se pokrene naredba (5) *Rsync* uslužni program provjerava stanje u folderu *izvjestaji_prod* na produkcijskom serveru i ukoliko u ovom folderu postoje datoteke koje se ne nalaze u folderu *izvjestaji_dr* na redundantnom serveru, tada će se te datoteke kreirati u folderu *izvjestaji_dr*. Takođe, ako neka datoteka postoji u folderu *izvjestaji_dr* na redundantnom serveru, a ne postoji u folderu *izvjestaji_prod* na primarnoj lokaciji, tada će prilikom pokretanja *Rsync* programa ta datoteka biti izbrisana na serveru na sekundarnoj lokaciji kako bi oba foldera sadržavala iste datoteke.

Brisanje datoteka u folderu na redundantnom serveru postižemo dodavanjem parametra *--delete* u naredbi (5). Parametar *--log-file=rsync_log* nam omogućava da se sve aktivnosti koje se izvrše pri pokretanju *Rsync* uslužnog programa zapišu u log fajl *rsync-log* na lokaciji */home/oracle* na produkcijskom serveru. Sinhronizacija se mora odvijati automatski, tj. *Rsync* uslužni program se neće pokretati ručno, nego će se automatski izvršavati u vrijeme koje je definisano u crontab datoteci. Cijeli proces se odvija tako što se u tačno određeno vrijeme pokreće *Rsync* uslužni program koji preko SSH protokola, koristeći ranije generisane ključeve, pristupa određišanom serveru.

C. Izvršavanje Rsync uslužnog programa kroz crontab

Posljednji korak u postizanju automatske sinhronizacije podataka između servera predstavlja podešavanje crontab datoteke. Pomoću nje će biti definisano tačno vrijeme kada će se pokretati *Rsync* koji je zadužen za sinhronizaciju. Zbog praktičnosti i preglednosti na produkcijskom serveru ćemo kreirati folder "skripte" u koji ćemo smjestiti *bash* skriptu *rsync_bckp.sh* koja će sadržavati naredbu (5), što je prikazano na Sl. 6.



Slika 6. Kreiranje bash skripte *rsync_bckp.sh*

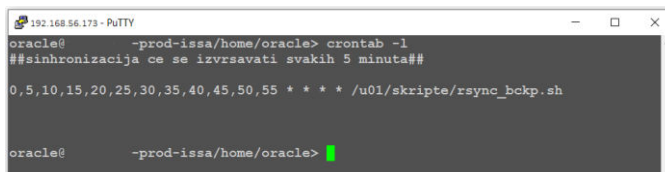
Nakon kreiranja *bash* skripte potrebno je omogućiti izvršavanje skripte, a to ćemo postići sljedećom naredbom:

```
chmod +x rsync_bckp.sh (6)
```

Sada kreiramo crontab datoteku, unosimo skriptu *rsync_bckp.sh* i podešavamo crontab tako da se skripta pokreće u intervalima od 5 minuta. Kroz komandni prozor naredbom:

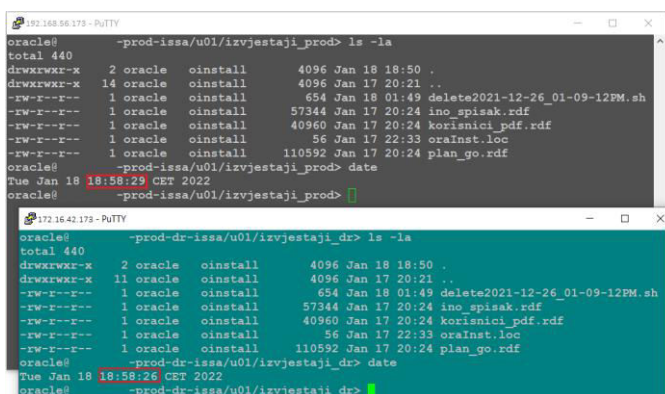
```
crontab -e (7)
```

definišemo vrijeme izvršavanja skripte *rsync_bckp.sh*, što je prikazano na Sl. 7.



Slika 7. Podešavanje crontab datoteke

Podešavanjem crontab datoteke naredbom (7) servis je aktivan u pozadini. Da bismo provjerili da li *crontab* radi onako kako želimo, prvo ćemo provjeriti koje datoteke se nalaze u folderima *izvjestaji_prod* i *izvjestaji_dr*. Provjeru smo uradili u 18:58 i utvrdili da se u oba foldera na serverima na produkcijskoj i sekundarnoj strani nalaze iste datoteke, kao što se vidi na Sl. 8.



Slika 8. Sadržaj foldera prije sinhronizacije podataka

Da bismo potvrdili da *crontab* radi onako kako smo ga podesili, u 19:02 iz foldera *izvjestaji_prod* na produkcijskom serveru ćemo izbrisati datoteku *oralnst.loc* i kreirati datoteku *known_hosts*. Na Sl. 9 vidimo da se nakon ažuriranja na produkcijskom serveru folder na primarnoj i sekundarnoj lokaciji razlikuju. Prema podešavanjima koja smo uradili u *crontab*-u nakon narednog izvršavanja skripte *rsync_bckp.sh*, koje bi se trebalo dogoditi u 19:05, stanje u oba foldera bi trebalo da bude isto, tj. oba foldera bi trebala da sadrže iste datoteke.

Slika 9. Izmijenjeni sadržaj foldera prije sinhronizacije

Na Sl. 10 prikazani su podaci u folderima nakon izvršenja *Rsync* programa u 19:05 i vidimo da je uspješno izvršena sinhronizacija podataka tako da folderi na produkcijskom i redundantnom serveru sadrže iste datoteke.

Slika 10. Sadržaj foldera nakon sinhronizacije podataka

Uvidom u log fajl *rsync_log* potvrđujemo da su evidentirane sve aktivnosti tokom izvršenja *Rsync* programa. Na Sl. 11 je prikazan sadržaj log fajla gdje se vide operacije koje su se desile u trenutku izvršenja skripte (vrijeme 19:05).

Slika 11. Sadržaj log fajla nakon sinhronizacije podataka

IV. ZAKLJUČAK

U ovom radu je pokazano da *Rsync* uslužni program uz korištenje *crontab*-a i SSH protokola predstavlja veoma efikasan alat za sinhronizaciju podataka i replikaciju datoteka između udaljenih servera. Korištenjem ovog uslužnog programa postiže se visok stepen automatizacije procedura za sinhronizaciju podataka i izradu rezervnih kopija. Kada se koristi uz SSH protokol, *Rsync* omogućava šifriranje i zaštitu podataka u toku prenosa podataka između udaljenih servera. Zahvaljujući algoritmu koji koristi, *Rsync* omogućava prenos samo onih podataka koji su izmijenjeni, što smanjuje količinu podataka koji se prenose i zahtijeva manji propusni opseg.

LITERATURA

- [1] W. Preston, "Backup and Recovery", O'Reilly Media, Inc, 2007
- [2] <https://linux.die.net/man/1/rsync>
- [3] <https://www.adminschoice.com/crontab-quick-reference>
- [4] J. Lewis, "Linux Utilities Cookbook", Packt Publishing, 2013
- [5] <https://www.ssh.com/academy/ssh/keygen>
- [6] D. Barrett, R. Silverman, R. Byrnes, "SSH, the Secure Shell: The Definitive Guide", O'Reilly Media, Second edition, 2005

ABSTRACT

This paper will explain the use of the RSYNC utility for replicating files and data between two application servers. These are servers located in different locations where Oracle Fusion Middleware is installed. At the primary location there is a production server, while at the secondary location there is a copy of it whose task is to ensure the disaster recovery. In addition to the Rsync utility, cron service and SSH protocol were used to exchange data between servers.

The use of the Rsync utility for data synchronization between servers

Marko Marinković