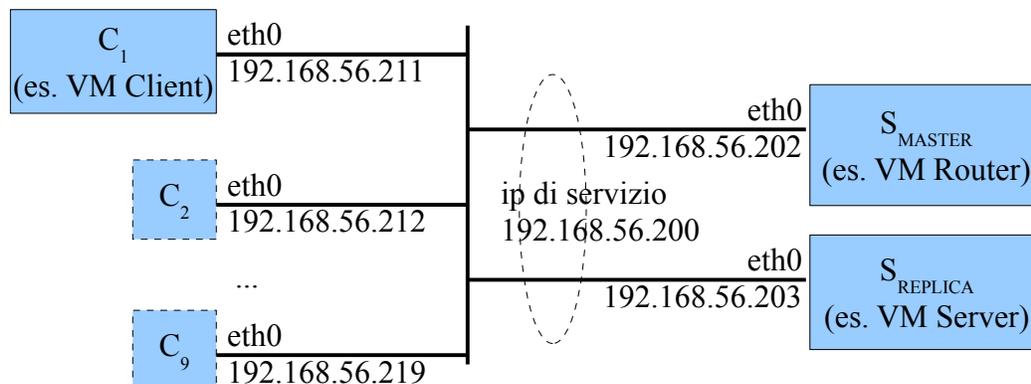


Laboratorio di Amministrazione di Sistemi T

Prova pratica - 1 luglio 2013

Descrizione generale del problema

Si consideri la rete illustrata in figura, in cui i blocchi C_i (per i compreso tra 1 e 9) rappresentano workstation disponibili all'utenza, collocate su di una rete privata su cui sono attestati anche due server (S_{MASTER} e $S_{REPLICA}$) che ospitano un sistema di monitoraggio ad alta disponibilità.



Il sistema controlla via SNMP lo stato di alcuni servizi che dovrebbero essere in esecuzione sui client, utilizzando la directory LDAP come database per i dati di configurazione lo storico delle misure, e registra sul log di sistema gli eventi rilevanti (le transizioni di stato dei servizi monitorati).

Per garantire il funzionamento in alta disponibilità del sistema, il servizio LDAP viene di norma erogato da S_{MASTER} , e i dati vengono periodicamente replicati su $S_{REPLICA}$.

Il servizio viene erogato utilizzando l'indirizzo 192.168.56.200 sull'alias $eth0:1$, che può essere configurato su $S_{REPLICA}$ se quest'ultimo rileva un malfunzionamento di S_{MASTER} .

I log vengono mantenuti sincroni in tempo reale tra i due server.

File da consegnare

stats.schema - Definire due classi:

- *monConfig* contiene gli attributi testuali *indirizzo* e *servizio*; una singola entry di questa classe per ogni client (identificato dall'indirizzo) conterrà tutti i nomi dei servizi (processi) da monitorare sul client stesso; (per fare i test durante l'esame, queste entry vanno inserite a mano).
- *monStatus* contiene gli attributi testuali *indirizzo*, *servizio* e *stato*; un'entry conserva lo stato ("on" oppure "off") del servizio rilevato sul client identificato dall'indirizzo.

init.sh - Questo script configura i server (rilevando automaticamente su quale dei due è in esecuzione)

- per bloccare tutto il traffico non strettamente necessario nel caso di funzionamento normale dei sistemi;
- per far sì che tutti i log etichettati `local1.info` di *S_{MASTER}* vengano sia scritti localmente sul file `/var/log/events`, sia inviati a *S_{REPLICA}* perché li scriva in un proprio file omonimo.

Indicare nei commenti come configurare i sistemi perché lo script sia eseguito al boot.

ha.sh - Questo script gira ininterrottamente su entrambi i server. Esegue ogni 5 secondi un controllo della situazione dei server, per soddisfare attraverso opportune invocazioni degli script *replica.sh*, *failover.sh* e *fencing.sh* la seguente politica:

- se entrambi i server sono attivi, il contenuto della directory LDAP di *S_{MASTER}* deve essere replicato su *S_{REPLICA}*;
- se *S_{MASTER}* non risponde alle query LDAP, *S_{REPLICA}* deve prenderne il posto;
- se *S_{MASTER}* si accorge che *S_{REPLICA}* ha iniziato ad erogare il servizio LDAP, deve rilasciare immediatamente l'IP di servizio e disattivare la directory LDAP locale.

Indicare nei commenti come garantire che lo script sia permanentemente in esecuzione.

replica.sh – Questo script accetta come parametro un indirizzo di un server LDAP, a cui si connette per scaricare tutte le entry di classe *monConfig* e *monStatus*; queste vanno a sostituire il contenuto della directory LDAP locale.

failover.sh – Questo script assegna l'IP di servizio alla macchina su cui è lanciato, ed apre il firewall per consentire le connessioni a LDAP dall'esterno.

fencing.sh – Questo script rilascia l'IP di servizio e disattiva la directory LDAP locale.

monitor.sh - Questo script controlla via SNMP tutti i client, verificando per ognuno lo stato dei servizi configurati nella corrispondente entry di classe *monConfig*; se rileva una variazione di stato rispetto all'ultimo controllo (il cui esito è memorizzato nella corrispondente entry di classe *monStatus*) registra l'evento attraverso il system logger con etichetta `local1.info` ed aggiorna la entry LDAP.

Indicare nei commenti come garantire che lo script sia eseguito ogni 2 minuti.

snmpdconf.sh – Questo script, lanciato su di un client, interroga LDAP e riconfigura l'agent snmp locale che consente la rilevazione dei servizi, come richiesta dallo script *monitor.sh*.