Final Project

Contact TAs: vegetable@csie.ntu.edu.tw

Report Submission

- Report Due Time: 2017/6/28 (Wed.) 23:59
- 頁數六面以內，以中文或英文撰寫皆可。
- 建議包含內容：題目敘述（背景介紹）、系統簡介、實作過程、遇到的困難、分工。以上項目僅供參考，可自行增減項目，內容越完整分數將越高。
- 各組只需其中一名組員繳交 **final_[studentID].pdf** (e.g. `final_bxx902xxx.pdf`) 至 ceiba。

Presentation

- 報告日期：2017/6/26（Mon.）
  - 上午場 9:10-12:10: SA 題目的組別報告
  - 下午場 13:30-16:30: NA 題目的組別報告
- 教室：R102
- 各組報告時間 10 分鐘 + Q&A 5 分鐘
- 評分包含：教授評分、助教評分、同半場的組別互評
- 每位同學需待滿所屬半場的全程。例如報告 SA 題目的組別，需出席上午場全程，下午場則自由參加。報告 NA 題目的組別相反。

Network Administration Projects

1 Network Switch Configuration Interface Library (libnsci)

- 利用 Python 3 實做 HP 與 Cisco switch 硬體抽象層 (HAL) 功能。
- 由於 HP 1820-8G switch 只有網頁管理界面，希望能與 Cisco CLI 界面整合，讓設定 Switch 時具有更高的統一性，都能使用相同的 Cisco CLI 指令。

1.1 器材

- HP 1820-8G Switch J9979A
- Cisco Catalyst 2960 Series Switch

1.2 目標

1. 透過相同的 Cisco 指令，對不同廠牌的 switch 做以下設定：
   (a) 更改 IP、switch 名稱、NTP server、local 端帳號密碼，設定 vlan、port channel，回傳 mac address table、孔位狀態、vlan membership、switch 狀況等常用功能。(目前 NA 小組已經初步實做完 CLI 控制 HP 網頁界面管理的部份 https://github.com/bookgin/hp1820-cli，還剩下整合進 library。)
   (b) 需具備良好的可擴充性/可用性，供開發者方便快速的加入新廠牌的 switch，或新增功能。
2. 抽象化所有設定、交換器資訊、連接埠資訊為統一格式的物件，方便其他程式串接 API。

1.3 mentor

資工三 林書瑾，contact: b03902078@csie.ntu.edu.tw

2 IoT 網路監控

2.1 器材

- Single-Board Computers * N (e.g., odroid, etc)

2.2 目標

1. Server-client
2. Client (SBC)
   (a) Deploy in the different environments (physical position, VLANs, subnets, etc)
   (b) Monitor the network status, and report it to the server
   - Data: reachability, quality (latency, throughput, etc.)
   - Target: client to server, client to client (all-pairs)
3. Server

(a) GUI to see the report (e.g., website)

(b) Alarm (email, etc.) on conditions like unreachable or reachable but the quality lower than a certain threshold

(c) Alternative to coding from scratch: integrate with Amazon AWS CloudWatch

- Push metrics to AWS with Python/Node.js/Java APIS

2.3 mentor

資工三 張庭瑋，contact: b03902081@csie.ntu.edu.tw

3 交換器紀錄蒐集

Use common log collection services (prefer ELK) to tap data from switches for further analysis and monitoring. Currently, we use Cacti combined with SNMP to achieve the job, but we'd like to move on to ELK to enable real-time analysis.

3.1 器材

偵測無線訊號的設備（擇一）

- Switch (SNMP capable) x 1
- Server x 1

3.2 目標

1. Set up an ELK stack server (https://www.elastic.co/webinars/introduction-elk-stack).
2. Enable SNMP on target switch.
3. Allow the ELK stack to sink log from the switch.
4. (Bonus) Perform basic analysis for the log, e.g., detecting source of the MAC flapping and sending e-mail alerts.

3.3 mentor

資工三 劉彥廷，contact: b03902036@csie.ntu.edu.tw

4 系館無線訊號強度分析

系上的無線網路設備這學期剛進行汰換工程，而無線組正在觀察其效果並進行調整中。因此這樣的地理分析對於無線組設計 AP 擺放的位置等等很有參考價值，也能對系上同學的無線網路使用產生正面影響。系上部分區域，如地下室，使用情況波動較大，因此可能影響訊號及真實傳輸速度。於是在這樣的特定區域設置訊號與速度追蹤設備，並在出現異常時即時回報，也是很重要的任務。

4.1 器材

偵測無線訊號的設備（擇一）

- 手機
- 筆電
- Arduino + Wi-Fi signal sensor

4.2 目標

1. 寫 script 或利用現有軟體偵測訊號強度及所在位置。
2. 與測量時的地理位置結合，做出系館 WiFi 強度地圖（密度越高越佳）。
3. (Bonus）即時性：能在固定地點自動追蹤無線訊號與傳輸速度，並且呈現結果。
   - 更新到地圖中。
   - 若有異常，即時通知網管人員。
4. (Bonus）歷史查詢：提供這些特定地點的訊號與速度歷史資訊查詢

4.3 mentor

資工三 林祐萱，contact: linamy85@gmail.com

5 高可靠性 pfSense 叢集

pfSense 是一個網路裡重要的節點，許多重要服務倚賴著他。數學告訴我們：兩台 pfSense 同時掛掉的機率遠小於一台掛掉的機率；所以請架設多台 pfSense 並實作備援機制。

5.1 器材

- 可以開多台虛擬機器的電腦

5.2 目標

1. 利用虛擬機器架設類似於系館的網路環境。
2. 實際設定多台 pfSense 的備援機制 (關鍵字: pfSense high availability)。
3. 當主要的 pfSense 無預警掛點時，次要的 pfSense 可以自動地取而代之，使得 pfSense 提供的服務不間斷。

5.3 影響

- pfSense 叢集會是未來網路架構的一環。此題目的實作結果將會為我們防火牆組吸收、使用與參考，你們的經驗與技術將會是系館網路建設的重要部分。

5.4 mentor

資工三 陳力，contact: b03902083@csie.ntu.edu.tw

6 DNS 管理介面與 DNSSEC

DNS 是基本且重要的服務，最常有需要的除了一個易於管理的介面以外，還有制定已久但不是很普及的 DNSSEC (Domain Name System Security Extensions)，DNSSEC 藉由公鑰密碼學以及 DNS 階層架構確保 DNS 紀錄的完整性、來源的可驗證性以及不存在紀錄的可驗證性。

6.1 器材

- 虛擬機器

6.2 目標

1. 架設 authoritative server，設定一些測試用記錄並設定 DNSSEC 紀錄。
2. 拒絕大量查詢的 IP 位址，並留下紀錄或進行通知。
3. 透過網頁介面修改 DNS 紀錄。
    - 支援常見的紀錄種類，包含 A、MX、CNAME、NS、PTR、TXT 等。
    - 修改時檢查紀錄是否正確 (例如字元是否符合規定)。
    - 修改後一併設定 DNSSEC 紀錄。
    - 權限區分，除了管理員以外各使用者只能修改屬於自己的紀錄，增加/刪除的權限和修改的權限分開。

6.3 mentor

資工三 楊松道，contact: b03902064@csie.ntu.edu.tw

System Administration Projects

1 系統、資源、與日誌之監控管理

今年三月的新聞，Elastic Stack 已經到了一個億的量級 [註 1]。Elastic Stack 又稱 ELK Stack，是非常火紅的開源日誌管理技術棧（Stack）。日誌的蒐集與管理對系統管理團隊（SA）的重要性自然不在話下；同時我們也希望能整合計算資源、系統資訊的監控，對於整個系統的運作能有快速且直接的認知。

1.1 器材

- 實體機 + Disk Array * 1

1.2 目標

1. 在實體機上建立若干 VM，部署 Elastic Stack。部署過程須用 Ansible 等自動化工具完成。
2. 日誌收集與儲存：透過 Logstash 等，蒐集並儲存各機器的 logs。
3. 機器狀態監控：透過 Nagios、Monit 等，監控各機器的狀態。
   (a) 監控磁碟狀態 (如 S.M.A.R.T 資訊)
      - 自動回報異常
   (b) 監控計算資源使用狀態
      - 負載高時自動 renice 部分 process
      - 需有 process 白名單及 user 白名單
   (c) 監控各項服務狀態
      - 異常時自動重啟必要的 daemon
   (d) 視覺化呈現：使用 Kibana 等，做資料視覺化呈現。

1.3 關鍵字

ELK Stack、Elastic Stack、日誌（Log）管理、大數據（Big Data）、系統監控（Monitoring）

1.4 備註

1. Elastic 公司 Elastic Stack 下載量高達 1 億次
2. 建设 DevOps 统一运维监控平台，先从日志监控说起
3. The Complete Guide to the ELK Stack
4. 不一定要使用題目中所提的套件，但請盡量使用開放原始碼套件與 Scripting Languages

1.5 mentor

資工四 鄭儒謙，contact: b02902001@csie.ntu.edu.tw

2 NFS 伺服器靜態負載平衡

2.1 器材

- VM × 3

2.2 目標

1. 你有 N 台 NFS 伺服器用來提供 M 位使用者工作站家目錄存取，該 M 位使用者以系級分配每 K 位一組，共有 U = M/K 組系級。請你設計一個 Hash function 把這 U 組平均（pseudo randomly）分配到 N 台 NFS 伺服器。請再架一台空機，上面有該 M 位使用者帳號。一開始所有使用者家目錄皆未掛載，當某使用者 X 登入時，請用 autofs 動態掛載其家目錄，找到 X 所屬系級的 NFS 伺服器位置方法可能有：

   (a) 已用 hash function 產生對應的 DNS 紀錄，放在 DNS 伺服器上。

   (b) 用 autofs executable map 來執行 hash function。

   (c) 已用 hash function 產生對應的紀錄，存在每一台 NFS 伺服器的 /etc/exports 中 refer=xxx 上。

2. 如果有新的系級加入，請用 script 自動在 Hash 相對應的 NFS 伺服器創該系級 K 位使用者的家目錄。

3. 家目錄權限可以設 777，擁有者不必是該使用者，因為這個任務並沒有要求統一伺服器和工作站的帳號。

4. 如果有新的 NFS 伺服器加入（假如系上買新硬體），你必須重新調整家目錄位置，把一些家目錄移到新的伺服器。作法可能是創一個新的 hash function，並有效的動態更新 DNS 或 autofs 設定檔。

5. 如果有 NFS 伺服器必須汰換，你也必須調整 hash function，並把上面的家目錄移到其他伺服器。

6. 因為動態無縫切換可能有點困難，你可以在更新 hash function 後把受影響的家目錄 umount 掉，再讓 autofs 自己 mount 到新的位置上（在現實生活中，就是通知那些使用者大概下線 5 分鐘左右）。

7. 這種負載平衡雖然 Glusterfs 可以輕易達成，但效能可能比較不好。你可以參考其實做。

2.3 mentor

資工三 陳耘志，contact: b03902074@csie.ntu.edu.tw

3 Arch Linux 工作站生成

3.1 器材

- VM × 3 - 4

3.2 目標

1. 架設一台 LDAP 伺服器、NFS 伺服器，分別提供工作站驗證服務與家目錄掛載，可能還需要一台 DNS 伺服器。
2. 需安裝所需所有套件，清單可見此。
3. 因為可能有多台工作站，對於 hostname 與 IP 分配須有完善規劃。
4. 用 Puppet 或 Ansible 等自動化工具**一鍵**生成工作站。
5. **Bonus**：切分內網 VLAN 和外網 VLAN，讓 LDAP、NFS 等服務在內網提供服務（以保護這些服務），外網則供使用者登入

3.3 mentor

資工三 陳耘志，contact: b03902074@csie.ntu.edu.tw

4 備援 (failover) web server

4.1 器材

- VM × 2 - 3

4.2 目標

1. 將同一個網站部署在兩個不同的機器上，當一台下線的時候，另一台能夠補上，並且通知你。
2. db 要能 sync，或是在壞掉之後再拿 db 的備份載入到另一台上。
3. 而當主要的 server 修復好了，可以手動將設定改回原樣，但必須用 script 的方式，讓人不用花太多時間操作。
4. 可以另開一台 vm 負責監控。

4.3 mentor

資工三 劉岳承，contact: b03902105@csie.ntu.edu.tw

5 信件伺服器

系上現有的信件服務有諸多可以改進的地方，未來改進的方向已經大致定好，然而改進以及未來維護需要的人力都不十分充足，希望能夠透過這個期末計畫讓有興趣的同學快速上手信件服務相關的知識，作為之後協助管理服務的人才。若簡單介紹信件服務，在收信的部份，可以大致上分為負責透過 SMTPs 協定接收信件的服務 Postfix ，以及儲存並讓使用者透過 IMAPs 協定存取信箱的 Dovecot 服務。由於一些因素使得 Dovecot 較不容易達到高可得性（High Availability，或是理解成備援）的目標，因此不是短期的目標。然而收信這件事的穩定性是十分重要的，因此未來規劃使用多個 Postfix 伺服器互相備援，對應到單一的 Dovecot 伺服器，這樣即使 Dovecot 暫時故障，遠方朋友寄來的信仍然可以被任一台 Postfix 伺服器接收並保留著，不會有單點失效的問題存在。關於 Postfix 和 Dovecot 更多的介紹，可以參考這個投影片。另外，未來還希望能夠讓服務的部署能夠使用 Ansible 達到全面自動化。Ansible 是一個讓你可以使用程式碼來管理伺服器的一套工具，這裡有一份簡單的教學。以下列出一些希望能達到的目標，有興趣的組別可以從裡面挑數個完成。

5.1 器材

- 數台 VM

5.2 目標 (完成愈多愈好，大致按照優先排序)

1. 使用安全的連線協議。
2. 一或數台可以收信的 Postfix Server。
3. 一台可以存信的 Dovecot Server。
4. 與 LDAP 或其他資料庫串接存放使用者資訊。
5. 使用 fail2ban 抵擋暴力猜密碼攻擊。
6. 使用 Ansible 自動化部署。
7. 多台 Postfix Server 已達到備援的效果。
8. 自動備份使用者信件
9. 使用 Postgrey, SpamAssassin, Amavis 或其他軟體過濾垃圾信件。
10. 架設 RoundCube, Horde5 或其他網路信件服務。
11. 架設寄信伺服器。

5.3 備註

1. 可能會提供具有系上私有 IP 的 VM
2. 可能會協助設置 DNS MX 紀錄。

5.4 mentor

資工三 江廷睿，contact: b03902072@csie.ntu.edu.tw

6 Fedora 工作站生成

系統管理團隊（SA）在歷經 2016 作業系統大戰之後，決定以 CentOS 為基底，統一管理用的 linux 發行版，著眼的好處是可持續性的教學與技能組（Skill Set）熟練。CentOS 當然不是銀彈，有適合它的場景，例如說需要長期穩定的服務，套件不需要新，但是要全；也有它力有未逮之處，例如說使用者想要最新的測試版本（Nightly Build），可能就沒有 RPM 的現成套件。相對於系統服務，如網站、LDAP…等，本系的 linux 工作站在使用者要求下，必須採用較新的套件，我們既想要可持續性的教學與技能組熟練，又想要嘗鮮，同樣是 RPM 套件管理的 Fedora 工作站看起來似乎是合理的選擇。工作站的發行版兩年前是 Debian ，後來換至自訂性較高的 Arch Linux，沿用至今。我們希望能做 Fedora 工作站的建置嘗試，確保 SA 在實體機、虛擬機的管理與佈署上，能夠充份利用共同採用 RPM 管理以及同屬 Red Hat 技術棧（Stack）的好處。

6.1 器材

- 至少一台實體機、一個 Public IP，及 Cobbler、Ansible、Gitlab…等基礎設施服務。

6.2 目標

自動化流水線（Pipeline）佈署至少一台 Fedora 工作站，作為基礎建設即代碼（Infrastructure as Code，IaC）的初步嘗試，必須安裝的套件可以參考樓上的 Arch Linux 工作站生成專案，或是 pkgbuild 。替換項目（技術上的地位接近或功能類似但實作不同）參考：

- Arch Linux <-> Fedora
- Aur <-> Rpm
- Trac <-> Gitlab
- Puppet <-> Ansible
- PXE <-> Cobbler

6.3 關鍵字

基礎建設即代碼（Infrastructure as Code，IaC）、紅帽（Red Hat）、Fedora、CentOS、Arch Linux、pkgbuild、rpm-build

6.4 mentor

網路/系統管理助教 吳一德，contact: wyde@csie.ntu.edu.tw

7 高可用性（High Availability，HA）集中帳號管理

系上 IT 服務的使用者包含校友，大概數千接近一萬這個量級，我們使用 OpenLDAP 來進行集中帳號管理。OpenLDAP 是一款羽量級目錄存取協定（Leightweight Directory Access Protocol，LDAP）的實現（以下直接以 LDAP 簡稱系上的 OpenLDAP 架設），預設以 Berkeley DB 作為後端資料庫，對於一次寫入資料、多次查詢和搜索有很好效果。當所有帳號與權限統一由一個系統管理，單點失效（Single Point of Failure）的影響就變得相當重要。我們可以透過硬體設備實現負載平衡及高可用性，例如 F5 或 A10 Networks的商用解決方案。也可以使用開源軟體實現 HA 架構，一個簡單的想法是透過 LDAP 服務的 IP 後面先擋一組主從架構（Master-Slave）的 LVS（Linux Virtual Server），再關聯到兩台一組 LDAP 伺服器。

7.1 器材

- Ansible VM * 1 (可與它項專案合用)
- LVS VM * 2
- LDAP VM * 2
- VIP * 1、內網 IP * 4
- Client VM * 1
- Client Web Service VM * 1

7.2 目標

用 Ansible 架設一組 LDAP HA 架構，在其中一台 LDAP 伺服器下線之後（例如 $ systemctl stop slapd），仍能維持帳號認證服務。這裡的挑戰是

1. LDAP 設定檔細節不少，更因為細節多，必須自動化管理，所以要用 Ansible 來部署。
2. LVS Cluster 有三種工作模式（NAT、DR、TUN），兩組 LDAP 伺服器有五種同步模式（Syncrepl、N-Way Multi-Master、MirrorMode、Syncrepl Proxy、Delta-syncrepl），DR + MirrorMode 看起來是不錯的組合，需要實測。
3. 部署 LVS、LDAP、Keepalived 都需要寫 script ，監測 VM 健康狀況也是。

7.3 關鍵字

OpenLDAP、高可用性（High Availability，HA）、單點失效（Single Point of Failure）

7.4 mentor

網路/系統管理助教 吳一德，contact: wyde@csie.ntu.edu.tw