Name: Renier L. Lope	**Date Performed: 09/30/2024**
Course/Section: CPE212 - CPE31S2	**Date Submitted: 10/02/2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem(2024-2025)**

Activity 6: Targeting Specific Nodes and Managing Services

1. **Objectives:**

1.1 Individualize hosts
1.2 Apply tags in selecting plays to run
1.3 Managing Services from remote servers using playbooks

2. **Discussion**:

In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are going to take a look at in this activity.

We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

Requirement:
In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installation. Take note of the IP address of Server 3. Make sure to use the command *ssh-copy-id* to copy the public key to Server 3. Verify if you can successfully SSH to Server 3.

Task 1: Targeting Specific Nodes

1. Create a new playbook and name it site.yml. Follow the commands as shown in the image below. Make sure to save the file and exit.

```
---

- hosts: all
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

```
 GNU nano 2.9.3                    site.yml

---

- hosts: all
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

2. Edit the inventory file. Remove the variables we put in our last activity and group according to the image shown below:

```
[web_servers]
192.168.56.120
192.168.56.121

[db_servers]
192.168.56.122


[file_servers]
192.168.56.123
```

Make sure to save the file and exit.

```
[web_servers]
server1
server2

[db_servers]
centOS

[file_servers]
server3
```

Right now, we have created groups in our inventory file and put each server in its own group. In other cases, you can have a server be a member of multiple groups, for example you have a test server that is also a web server.

3. Edit the *site.yml* by following the image below:

```
---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"


- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

Make sure to save the file and exit.

The *pre-tasks* command tells the ansible to run it before any other thing. In the *pre-tasks*, CentOS will install updates while Ubuntu will upgrade its distribution package. This will run before running the second play, which is targeted at *web_servers*. In the second play, apache and php will be installed on both Ubuntu servers and CentOS servers.

Run the *site.yml* file and describe the result.

```
PLAY RECAP *********************************************************************
*
centOS                     : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server2                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server3                    : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

- **It runs all tasks perfectly.**

4. Let's try to edit again the *site.yml* file. This time, we are going to add plays targeting the other servers. This time we target the *db_servers* by adding it on the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.

```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

  - name: install mariadb packege (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

Make sure to save the file and exit.
Run the *site.yml* file and describe the result.

```
PLAY RECAP *********************************************************************
*
centOS                     : ok=5    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server2                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server3                    : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

- **It runs all the plays perfectly including the plays targeting db_servers.**

5. Go to the remote server (Ubuntu) terminal that belongs to the db_servers group and check the status for mariadb installation using the command: *systemctl status mariadb.* Do this on the CentOS server also.
   Describe the output.

```
[rnrlope@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor pres
et: disabled)
   Active: active (running) since Sun 2024-09-29 22:03:40 EDT; 58s ago
  Process: 22249 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=ex
ited, status=0/SUCCESS)
  Process: 22161 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exite
d, status=0/SUCCESS)
 Main PID: 22248 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─22248 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─22414 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysq...

Sep 29 22:03:36 localhost.localdomain mariadb-prepare-db-dir[22161]: MySQL ma...
Sep 29 22:03:36 localhost.localdomain mariadb-prepare-db-dir[22161]: Please r...
Sep 29 22:03:36 localhost.localdomain mariadb-prepare-db-dir[22161]: The late...
Sep 29 22:03:36 localhost.localdomain mariadb-prepare-db-dir[22161]: You can ...
Sep 29 22:03:36 localhost.localdomain mariadb-prepare-db-dir[22161]: http://d...
Sep 29 22:03:36 localhost.localdomain mariadb-prepare-db-dir[22161]: Consider...
Sep 29 22:03:36 localhost.localdomain mariadb-prepare-db-dir[22161]: https://...
Sep 29 22:03:36 localhost.localdomain mysqld_safe[22248]: 240929 22:03:36 mys...
```

   - **It displays that mariadb.service is active(running).**

6. Edit the *site.yml* again. This time we will append the code to configure installation on the *file_servers* group. We can add the following on our file.

```
- hosts: file_servers
  become: true
  tasks:

  - name: install samba package
    package:
      name: samba
      state: latest
```

   Make sure to save the file and exit.
   Run the *site.yml* file and describe the result.

```
TASK [install samba package] ***************************************************
*
changed: [server3]

PLAY RECAP *********************************************************************
*
centOS                     : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server2                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server3                    : ok=4    changed=1    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

   - **It runs all the plays perfectly including the plays targeting file_servers.**

The testing of the *file_servers* is beyond the scope of this activity, and as well as our topics and objectives. However, in this activity we were able to show that we can target hosts or servers using grouping in ansible playbooks.

Task 2: Using Tags in running playbooks

In this task, our goal is to add metadata to our plays so that we can only run the plays that we want to run, and not all the plays in our playbook.

1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.

```
---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    tags: always
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"
```

```
- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    tags: apache,apache2,ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    tags: apache,centos,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    tags: centos, db,mariadb
    dnf:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

  - name: install mariadb packege (Ubuntu)
    tags: db, mariadb,ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"

- hosts: file_servers
  become: true
  tasks:

  - name: install samba package
    tags: samba
    package:
      name: samba
      state: latest
```

Make sure to save the file and exit.

Run the *site.yml* file and describe the result.

```
PLAY RECAP *********************************************************************
*********************************************************************
centOS                     : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server2                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
rnrlope@workstation:~/CPE212_LOPE_Act6$
```

- **It runs perfectly, I didn't see any changes with the results compared to the one without tags.**

2. On the local machine, try to issue the following commands and describe each result:

2.1 *ansible-playbook --list-tags site.yml*

- ***Issuing this command displays the tags assigned to each play inside a playbook.***

```
rnrlope@workstation:~/CPE212_LOPE_Act6$ ansible-playbook --list-tags site.yml

playbook: site.yml

  play #1 (all): all    TAGS: []
      TASK TAGS: [always]

  play #2 (web_servers): web_servers    TAGS: []
      TASK TAGS: [apache, apache2, centos, httpd, ubuntu]

  play #3 (db_servers): db_servers      TAGS: []
      TASK TAGS: [centos, db, mariadb, ubuntu]

  play #4 (file_servers): file_servers  TAGS: []
      TASK TAGS: [samba]
rnrlope@workstation:~/CPE212_LOPE_Act6$
```

2.2 *ansible-playbook --tags centos --ask-become-pass site.yml*

- ***Issuing this command will run only the tasks tagged with "centos" inside the playbook.***

```
TASK [Gathering Facts] ****************************************************
*
ok: [centOS]

TASK [install mariadb package (CentOS)] ***********************************
*
ok: [centOS]

PLAY [file_servers] *******************************************************
*

TASK [Gathering Facts] ****************************************************
*
ok: [server3]

PLAY RECAP ****************************************************************
*
centOS                     : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
server1                    : ok=3    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server2                    : ok=3    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server3                    : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

2.3 *ansible-playbook --tags db --ask-become-pass site.yml*

- ***This command will only run tasks tagged with "db" inside the playbook.***

```
TASK [install mariadb package (CentOS)] ************************************
*
ok: [centOS]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [centOS]

PLAY [file_servers] ********************************************************
*

TASK [Gathering Facts] *****************************************************
*
ok: [server3]

PLAY RECAP *****************************************************************
*
centOS                     : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server1                    : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
server2                    : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
server3                    : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

2.4 *ansible-playbook --tags apache --ask-become-pass site.yml*

- ***This command only runs task tagged with "apache" inside the playbook***

```
TASK [Gathering Facts] *****************************************************
*
ok: [centOS]

PLAY [file_servers] ********************************************************
*

TASK [Gathering Facts] *****************************************************
*
ok: [server3]

PLAY RECAP *****************************************************************
*
centOS                     : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server2                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server3                    : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

2.5 *ansible-playbook --tags “apache,db” --ask-become-pass site.yml*

- ***Issuing this command will run only the tasks tagged either with “apache” or “db” inside the playbook.***

```
TASK [install mariadb package (CentOS)] ************************************
*
ok: [centOS]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [centOS]

PLAY [file_servers] ********************************************************
*

TASK [Gathering Facts] *****************************************************
*
ok: [server3]

PLAY RECAP *****************************************************************
*
centOS                     : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server2                    : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
server3                    : ok=3    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

Task 3: Managing Services

1. Edit the file site.yml and add a play that will automatically start the httpd on CentOS server.

```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- name: start httpd (CentOS)
  tags: apache, centos,httpd
  service:
    name: httpd
    state: started
  when: ansible_distribution == "CentOS"
```

Figure 3.1.1

Make sure to save the file and exit.

```
- name: start httpd (CentOS)
  tags: apache, centos,httpd
  service:
    name: httpd
    state: started
  when: ansible_distribution == "CentOS"
```

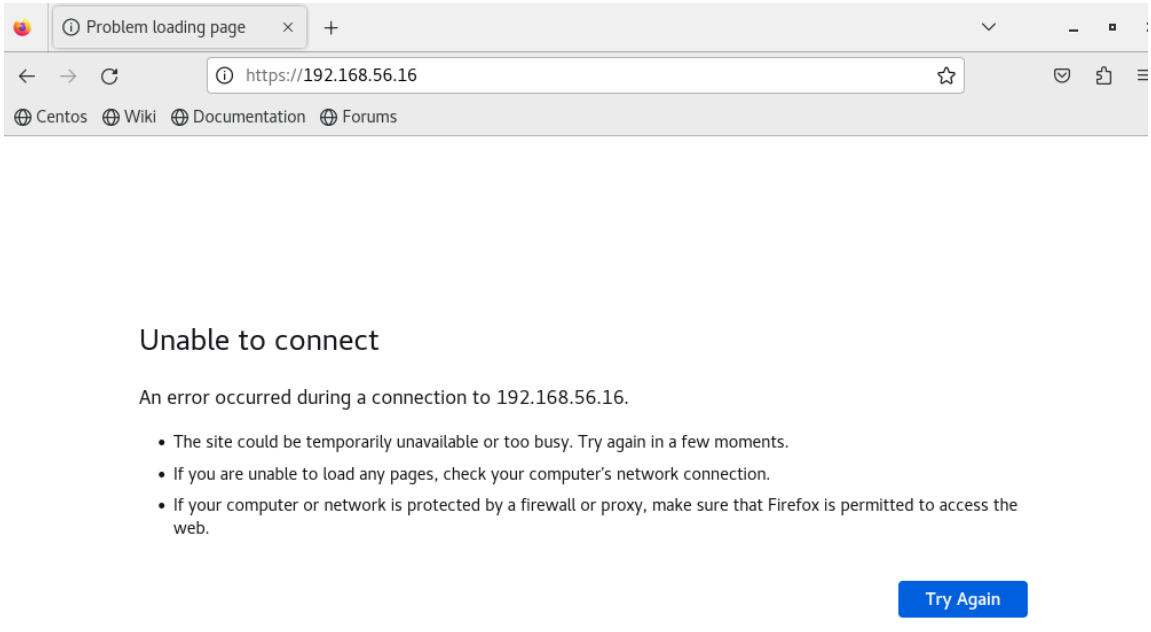
You would also notice from our previous activity that we already created a module that runs a service.

```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    tags: centos, db,mariadb
    dnf:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true
```

Figure 3.1.2

This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd.* When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display because we stopped the httpd service already.



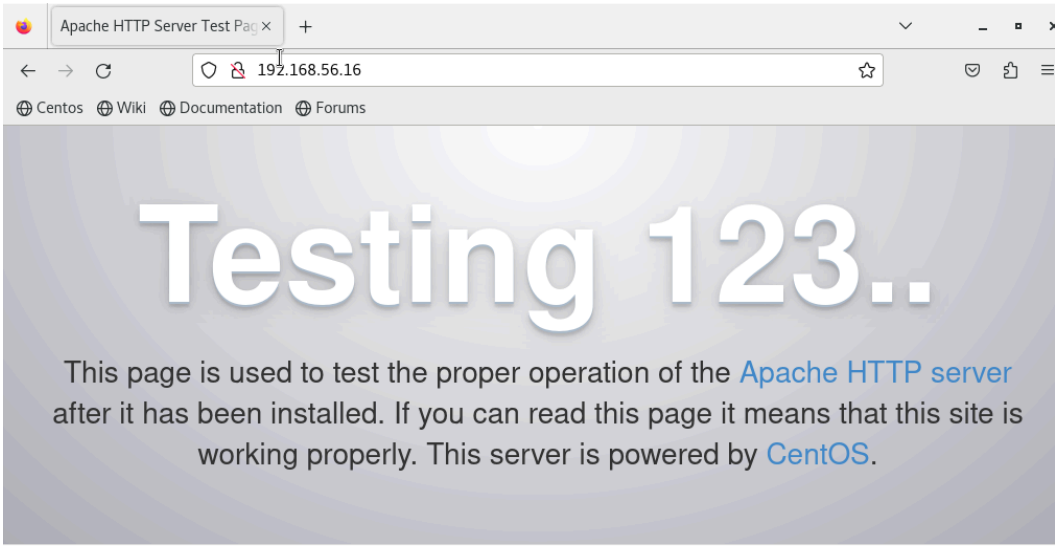
3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.
    - **The Task "start httpd (CentOS)" changed something in CentOS, which means it runs perfectly.**

note: I added the db_server to the play, because my CentOS is my db_server.

```
TASK [start httpd (CentOS)] ************
*
skipping: [server1]
skipping: [server2]
changed: [centOS]
```

```
PLAY RECAP *********************************************************************
*
centOS                     : ok=8    changed=2    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server1                    : ok=4    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server2                    : ok=4    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server3                    : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act6$
```

To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.




```
[rnrlope@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: active (running) since Tue 2024-10-01 20:58:26 EDT; 4min 52s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 10554 (httpd)
```

GIT PUSH:

```
rnrlope@workstation:~/CPE212_LOPE_Act6$ git add site.yml
rnrlope@workstation:~/CPE212_LOPE_Act6$ git add inventory
rnrlope@workstation:~/CPE212_LOPE_Act6$ git add ansible.cfg
rnrlope@workstation:~/CPE212_LOPE_Act6$ git commit -m "Activity 6"
[main 1204425] Activity 6
 3 files changed, 3 deletions(-)
rnrlope@workstation:~/CPE212_LOPE_Act6$ git push origin main
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 554 bytes | 554.00 KiB/s, done.
Total 5 (delta 1), reused 1 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:RenierCode/CPE212_LOPE_Act6.git
   5ab70e5..1204425  main -> main
rnrlope@workstation:~/CPE212_LOPE_Act6$
```

GITHUB LINK: https://github.com/RenierCode/CPE212_LOPE_Act6.git

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   - **The importance of placing our remote servers into groups is that we can call them efficiently and assign them into different plays more easily instead of calling them one by one. It is incredibly important if you are handling a huge amount of nodes.**
2. What is the importance of tags in playbooks?
   - **Tags are important in playbooks because we can run a specific task inside the playbook using a specific tag instead of running all of the tasks inside the playbook which may take a long time.**
3. Why do I think some services need to be managed automatically in playbooks?
   - **Some services needed to be managed automatically in playbooks such as starting apache or enabling mariadb, so that it can relieve us of the hassle of manually starting and enabling them.**