LETTERKENNY INSTITUTE OF TECHNOLOGY

ASSIGNMENT COVER SHEET

Lecturer's Name: Ruth Lennon

Assessment Title: Scripting The Deploy Pipeline

Work to be submitted to: Ruth Lennon

Date for submission of work: 21/03/2022

Place and time for submitting work: Blackboard as per submission link

To be completed by the Student

Student's Name: Luis Gonzalez

Class: Scripting The Deploy Pipeline

Subject/Module: Scripting The Deploy Pipeline

Word Count (where applicable): N/A

I confirm that the work submitted has been produced solely through my own efforts.

Student's signature: Luis Gonzalez **Date:** 21/03/2022

Notes

Penalties: The total marks available for an assessment is reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. Assessment work received more than two weeks late will receive a mark of zero. [Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute's Assessment Regulations.]

Plagiarism: Presenting the ideas etc. of someone else without proper acknowledgement (see section L1 paragraph 8).

Cheating: The use of unauthorised material in a test, exam etc., unauthorised access to test matter, unauthorised collusion, dishonest behaviour in respect of assessments, and deliberate plagiarism (see section L1 paragraph 8).

Continuous Assessment: For students repeating an examination, marks awarded for continuous assessment, shall normally be carried forward from the original examination to the repeat examination.

Aims/Description

As per Assignment Question:

1. Create a web page with your Lnumber, name, date and time. Upload to a github repo. Set up a playbook to configure a WebServer
2. Set up a playbook to configure a Secure Database Server of your choice.
3. Only use the Ansible guidebook to help you. Copy a file from s3 to your database using Ansible.
4. Compare Ansible with other solutions you have used (Python, Terraform, CloudFormation) for the purposes of server configuration. Do not exceed 2 pages.
    a. Draw on your personal experience with each technology.
    b. Back up your opinion with referenced work.
    c. Images and references are not included in the page limit.

Results

1. VM's with Fedora server created for ansible lab successfully.
2. User configured ssh access with rsa_key and nopassword(sudoers).
3. Ansible and required tools installed in Ansible Control PC successfully.
4. Playbook created and tested to configure web server with packages, services and deploy website from github that communicates to database.
5. Playbook to download source file from AWS S3 and import to database created and tested successfully.
6. After running playbooks user was able to remote to target pc with ssh.
7. After running playbooks user was able to connect to database through db gui.
8. After running playbooks user was able to access website showing records in database.
9. Current file was uploaded to Student's GitHub Repository into Ansible https://github.com/L00170299/ScriptingTheDeployPipeline

Conclusions

Ansible is very easy to understand tool to configure and administrate resources (orchestration) Student found it super interesting and extremely easy to get started with. There is a huge list of modules than can be installed depending the tasks that are needed which makes it very dynamic and growing constantly.

Is very useful to automate repetitive tasks either in a single device or multiple devices that required the same set of tasks. Task like update operative system, creating files, download, upload, connect, etc. are the kind of task that can be done manually in a console or maybe can be automated using python or even using Cloud Formation. But each of these tools were created for different purposes.

Automate task is possible in so many ways and with a long list of tools. But each of those were created with a set of capabilities thought for a type of task in particular. Just because a tool is able to execute particular tasks doesn't mean is the correct tool or way of doing it.

For example, with python user could create functions to connect, download and install packages which is a completely valid use of it but if there is a tool that does the same with much less effort then we can save time and could guarantee that implementation and use of it is less painful if we considered that there is a long list of people/companies testing it.

The same way if I'm trying to create a infrastructure probably I would use cloud formation or vagrant which are tools created for that purpose event though python or Ansible would be able to automate some scripts here and there to achieve the same results. But again we would be "using/forcing" a tool to something that it wasn't designed for.

As part of course student has learned about new tools and has better idea of the best uses for tools. If student had to create new servers, configure, install pkgs, schedule repetitive tasks, create and deploy in house software this is how it probably would look like:

Cloud Formation/Terraform/Vagrant => create servers, networks, open ports.
Ansible => to install, update OS, run specific commands, maintenance, etc.
create tools (scripts, website, etc) => Python, PowerShell, C#, bash, etc.

The previous is just an attempt to exemplify which tool use for which task but there is tons of information to consider.

In an article by (Brikman, 2016) the author explains what are the factors they have considered to take a decisions between tools. I think those can be apply to different set of tools like if its open source, how easy is to implement, security, maintenance, the community, mature vs cutting edge, etc.

References & Bibliography

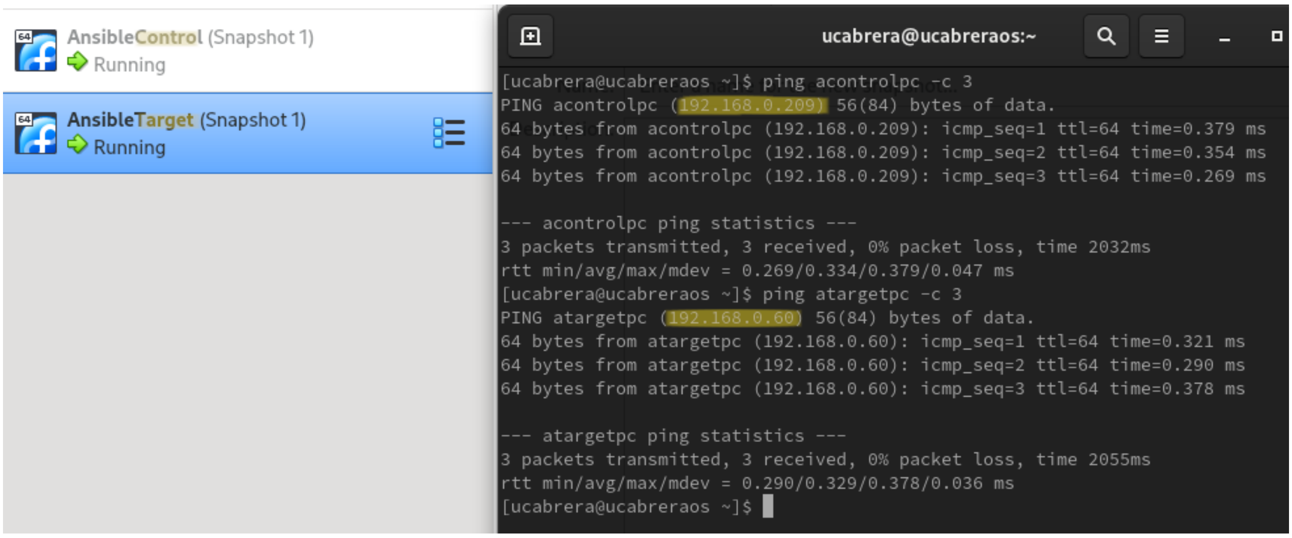
Brikman, Yevgeniy. (2016) "Why we use Terraform and not Chef, Puppet, Ansible, SaltStack, or CloudFormation".
Available at: https://blog.gruntwork.io/why-we-use-terraform-and-not-chef-puppet-ansible-saltstack-or-cloudformation-7989dad2865c (Accessed: 21 March 2022).

Terra, John. (2022) "Ansible vs Chef: What's the Difference?".
Available at: https://www.simplilearn.com/ansible-vs-chef-differences-article (Accessed: 21 March 2022).

Appendix

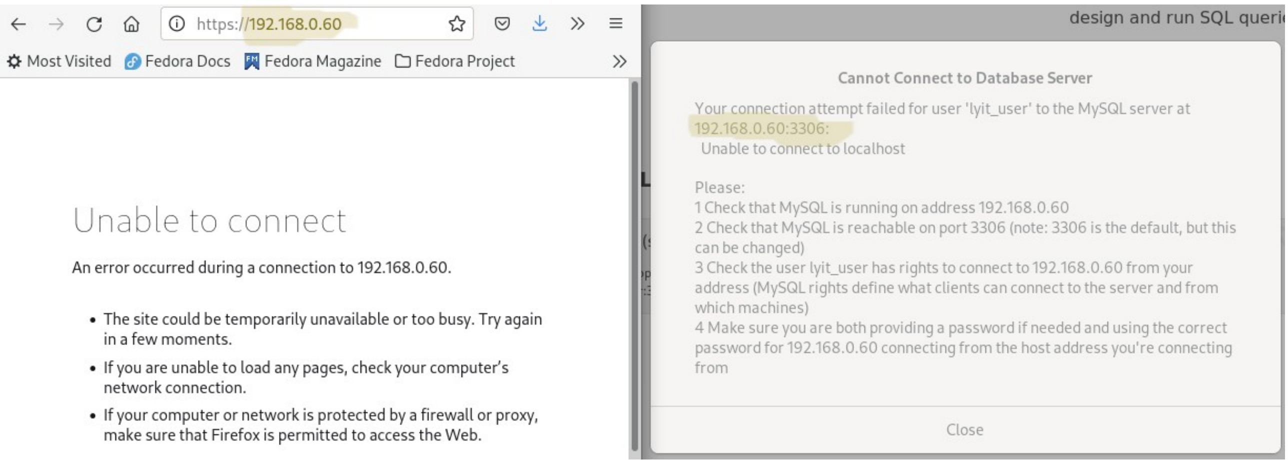
Clean VM’s with Fedora Server.
Devices:

acontrolpc
atargetpc



```
[ucabrera@ucabreraos ~]$ ping acontrolpc -c 3
PING acontrolpc (192.168.0.209) 56(84) bytes of data.
64 bytes from acontrolpc (192.168.0.209): icmp_seq=1 ttl=64 time=0.379 ms
64 bytes from acontrolpc (192.168.0.209): icmp_seq=2 ttl=64 time=0.354 ms
64 bytes from acontrolpc (192.168.0.209): icmp_seq=3 ttl=64 time=0.269 ms

--- acontrolpc ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.269/0.334/0.379/0.047 ms
[ucabrera@ucabreraos ~]$ ping atargetpc -c 3
PING atargetpc (192.168.0.60) 56(84) bytes of data.
64 bytes from atargetpc (192.168.0.60): icmp_seq=1 ttl=64 time=0.321 ms
64 bytes from atargetpc (192.168.0.60): icmp_seq=2 ttl=64 time=0.290 ms
64 bytes from atargetpc (192.168.0.60): icmp_seq=3 ttl=64 time=0.378 ms

--- atargetpc ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.290/0.329/0.378/0.036 ms
[ucabrera@ucabreraos ~]$ 
```

Target PC without webservice and not mysql



Unable to connect

An error occurred during a connection to 192.168.0.60.

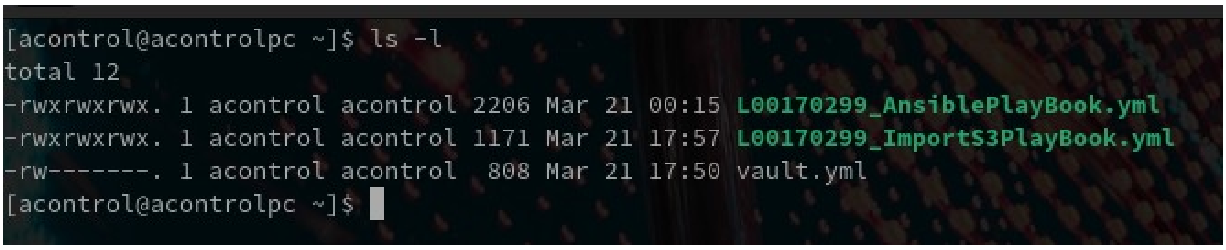
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer’s network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Cannot Connect to Database Server

Your connection attempt failed for user 'lyit_user' to the MySQL server at 192.168.0.60:3306:
Unable to connect to localhost

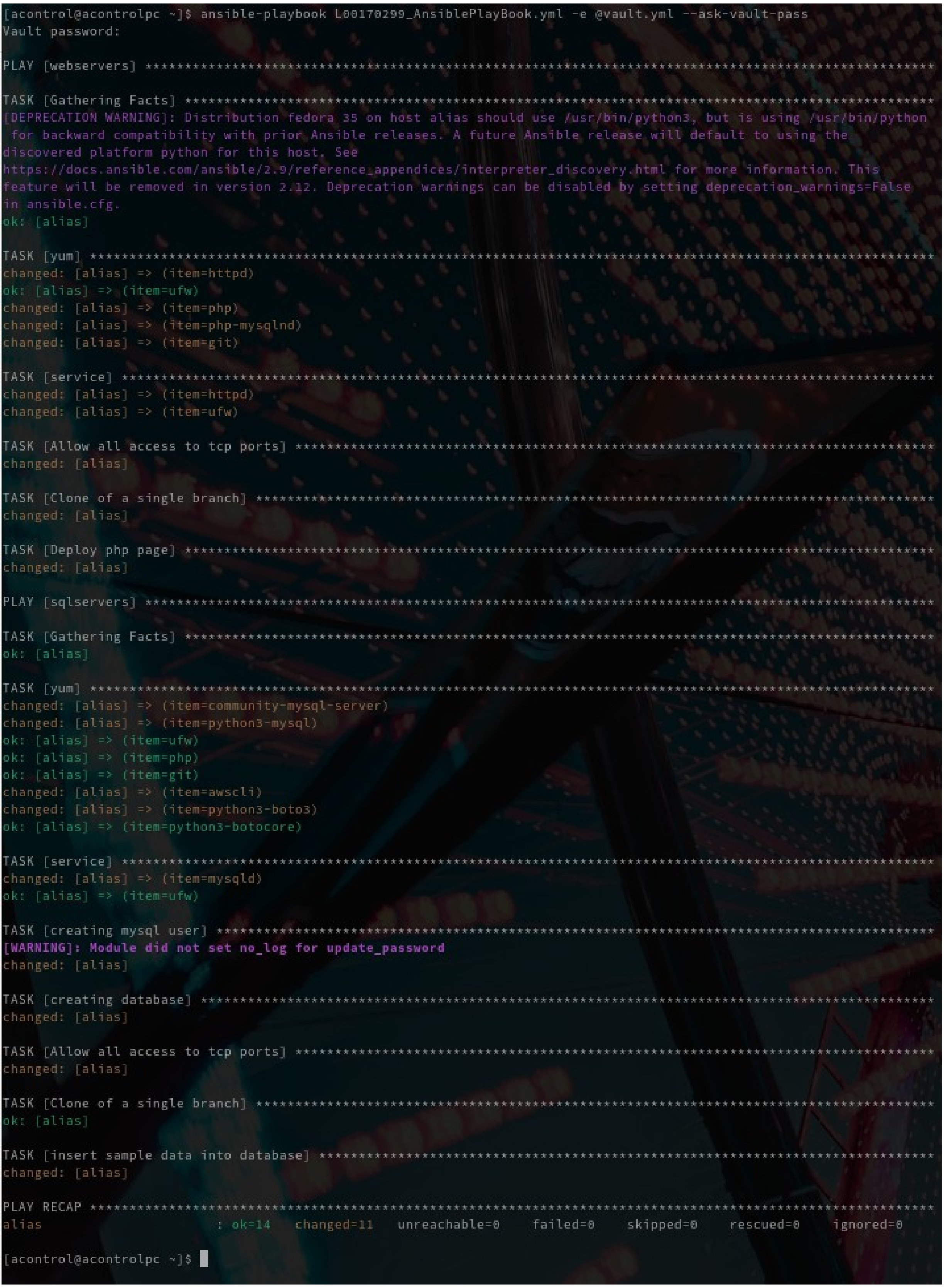
Please:
1 Check that MySQL is running on address 192.168.0.60
2 Check that MySQL is reachable on port 3306 (note: 3306 is the default, but this can be changed)
3 Check the user lyit_user has rights to connect to 192.168.0.60 from your address (MySQL rights define what clients can connect to the server and from which machines)
4 Make sure you are both providing a password if needed and using the correct password for 192.168.0.60 connecting from the host address you're connecting from

Control PC with 2 playbooks and a vault file to keep secrets.



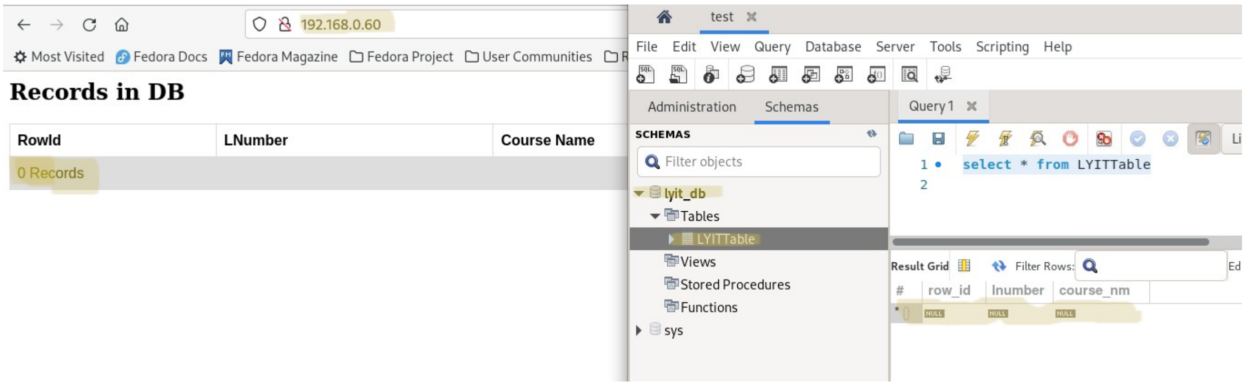
```
[acontrol@acontrolpc ~]$ ls -l
total 12
-rwxrwxrwx. 1 acontrol acontrol 2206 Mar 21 00:15 L00170299_AnsiblePlayBook.yml
-rwxrwxrwx. 1 acontrol acontrol 1171 Mar 21 17:57 L00170299_ImportS3PlayBook.yml
-rw-------. 1 acontrol acontrol  808 Mar 21 17:50 vault.yml
[acontrol@acontrolpc ~]$ 
```

Output from first playbook to configure webserver and sqlserver.



```
[acontrol@acontrolpc ~]$ ansible-playbook L00170299_AnsiblePlayBook.yml -e @vault.yml --ask-vault-pass
Vault password:

PLAY [webservers] ****************************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************
[DEPRECATION WARNING]: Distribution fedora 35 on host alias should use /usr/bin/python3, but is using /usr/bin/python
 for backward compatibility with prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False
in ansible.cfg.
ok: [alias]

TASK [yum] ***********************************************************************************************************
changed: [alias] => (item=httpd)
ok: [alias] => (item=ufw)
changed: [alias] => (item=php)
changed: [alias] => (item=php-mysqlnd)
changed: [alias] => (item=git)

TASK [service] *******************************************************************************************************
changed: [alias] => (item=httpd)
changed: [alias] => (item=ufw)

TASK [Allow all access to tcp ports] *********************************************************************************
changed: [alias]

TASK [Clone of a single branch] **************************************************************************************
changed: [alias]

TASK [Deploy php page] ***********************************************************************************************
changed: [alias]

PLAY [sqlservers] ****************************************************************************************************

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

TASK [yum] ***********************************************************************************************************
changed: [alias] => (item=community-mysql-server)
changed: [alias] => (item=python3-mysql)
ok: [alias] => (item=ufw)
ok: [alias] => (item=php)
ok: [alias] => (item=git)
changed: [alias] => (item=awscli)
changed: [alias] => (item=python3-boto3)
ok: [alias] => (item=python3-botocore)

TASK [service] *******************************************************************************************************
changed: [alias] => (item=mysqld)
ok: [alias] => (item=ufw)

TASK [creating mysql user] *******************************************************************************************
[WARNING]: Module did not set no_log for update_password
changed: [alias]

TASK [creating database] *********************************************************************************************
changed: [alias]

TASK [Allow all access to tcp ports] *********************************************************************************
changed: [alias]

TASK [Clone of a single branch] **************************************************************************************
ok: [alias]

TASK [insert sample data into database] ******************************************************************************
changed: [alias]

PLAY RECAP ***********************************************************************************************************
alias                      : ok=14   changed=11   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[acontrol@acontrolpc ~]$ 
```

Result after running first playbook to configure with website and database.



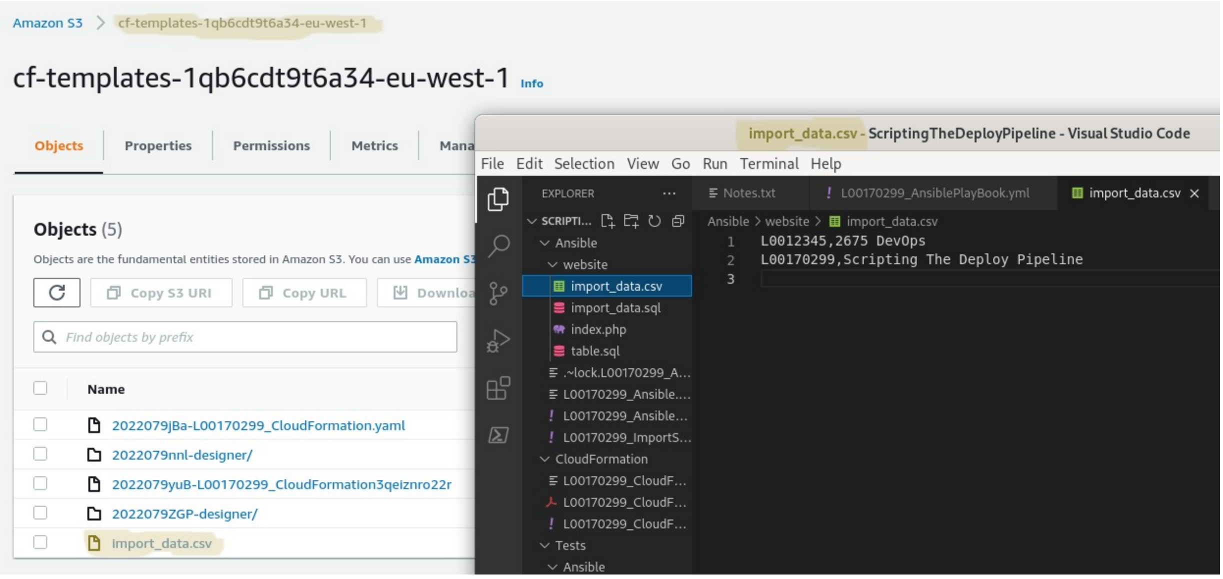
Records in DB

RowId	LNumber	Course Name
0 Records		

```
select * from LYITTable
```

#	row_id	lnumber	course_nm
	NULL	NULL	NULL

Source file and content of it in AWS S3 before running Importing playbook



cf-templates-1qb6cdt9t6a34-eu-west-1

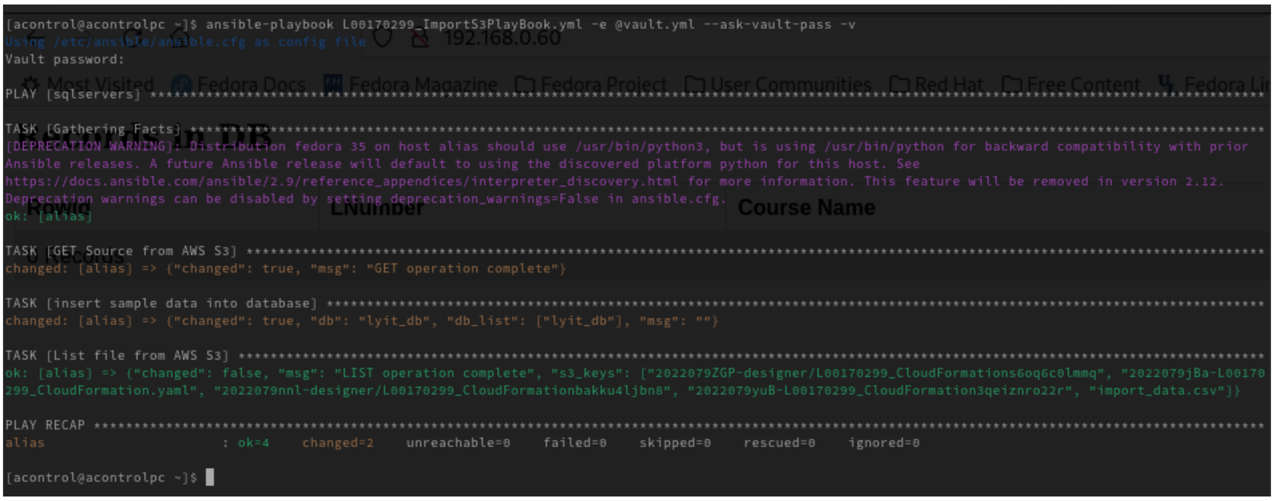
Objects (5)

Objects are the fundamental entities stored in Amazon S3. You can use **Amazon S3**

□	Name
□	2022079jBa-L00170299_CloudFormation.yaml
□	2022079nnl-designer/
□	2022079yuB-L00170299_CloudFormation3qeiznro22r
□	2022079ZGP-designer/
□	import_data.csv

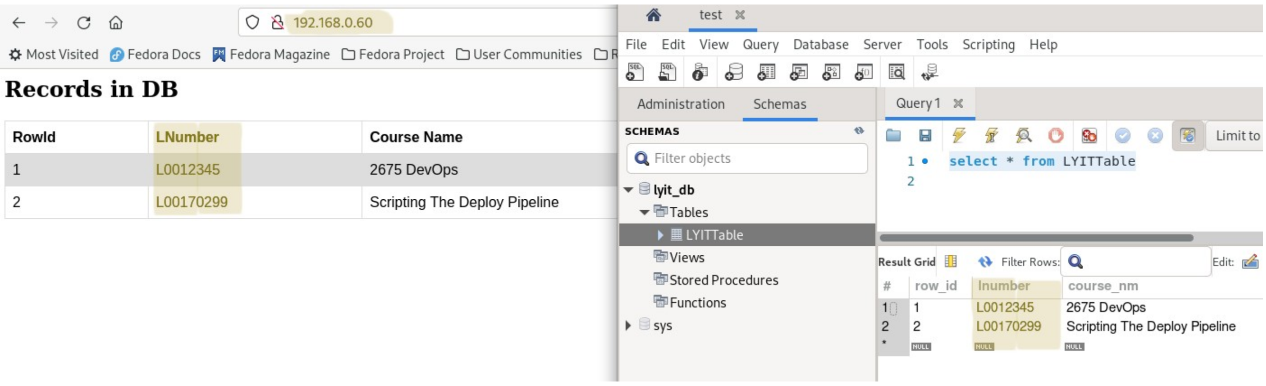
```
L0012345,2675 DevOps
L00170299,Scripting The Deploy Pipeline
```

Output from second playbook to download and import file from AWS S3



```
[acontrol@acontrolpc ~]$ ansible-playbook L00170299_ImportS3PlayBook.yml -e @vault.yml --ask-vault-pass -v
Using /etc/ansible/ansible.cfg as config file
Vault password:

PLAY [sqlservers] *************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
[DEPRECATION WARNING]: Distribution fedora 35 on host alias should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior
Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12.
Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [alias]

TASK [GET Source from AWS S3] *************************************************************************************************
changed: [alias] => {"changed": true, "msg": "GET operation complete"}

TASK [insert sample data into database] ***************************************************************************************
changed: [alias] => {"changed": true, "db": "lyit_db", "db_list": ["lyit_db"], "msg": ""}

TASK [List file from AWS S3] **************************************************************************************************
ok: [alias] => {"changed": false, "msg": "LIST operation complete", "s3_keys": ["2022079ZGP-designer/L00170299_CloudFormations6oq6c0lmmq", "2022079jBa-L00170
299_CloudFormation.yaml", "2022079nnl-designer/L00170299_CloudFormationbakku4ljbn8", "2022079yuB-L00170299_CloudFormation3qeiznro22r", "import_data.csv"]}

PLAY RECAP ********************************************************************************************************************
alias                      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[acontrol@acontrolpc ~]$ 
```

Result after running importing playbook with records in database and showing in website



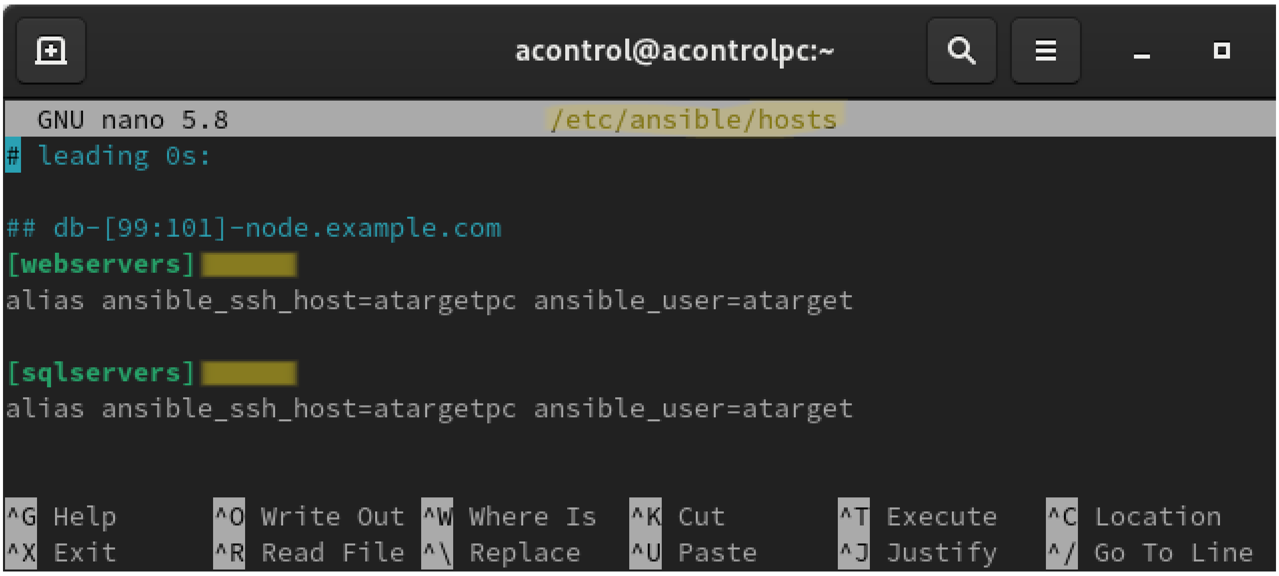
Records in DB

RowId	LNumber	Course Name
1	L0012345	2675 DevOps
2	L00170299	Scripting The Deploy Pipeline

```
select * from LYITTable
```

#	row_id	lnumber	course_nm
1	1	L0012345	2675 DevOps
2	2	L00170299	Scripting The Deploy Pipeline
*	NULL	NULL	NULL

Ansible Inventory with groups



```
GNU nano 5.8                /etc/ansible/hosts
# leading 0s:

## db-[99:101]-node.example.com
[webservers]
alias ansible_ssh_host=atargetpc ansible_user=atarget

[sqlservers]
alias ansible_ssh_host=atargetpc ansible_user=atarget


^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```