Task 2

Create a transition plan for upgrading servers with no to minimal downtime

1. For this assignment, write a plan for transitioning production servers from Centos6 + PHP5.5 + MySQL 5.5 to Centos7 + PHP7 + MySQL 5.7.

2. You don't need to perform the transition or write any code, just write up how you would solve the task.

3. The people reading your plan will have strong technical knowledge.

4. You can make any assumptions necessary to fill in the gaps, but for this assignment assume there is a 1 load balancer, 5 web servers, and 3 database servers.

5. In your plan, you can include anything you think is important, some suggestions include Assumptions, Schedule, Security, Testing, Requirements, Backups.

6. If you have questions please send an email to khalid.amin@edu2x.com with the subject "Devops Interview Question"

7. Deliverables

   1. Commit your plan to the branch: task_server_transition_plan

Answer

My Plan:

Let’s Divide my plan to 3 points:

1.Upgrade the OS from “Centos 6” to “Centos 7”

2.Upgrade the Mysql Package from “5.5” to “5.7”

3.Upgrade the PHP Package from “5.5” to “7”

Note:

i will explain every point through two sides “My search result” and “My experience”.

I will Assume that we have two infrastructure cases one of them on premise and the another hosted AWS.

Infrastructure Overview:

We have 9 Servers, 1 load balancer, 5 web servers, and 3 database servers.

1.I will start upgrading the OS for every server.

2.Then upgrading our services packages.

1.Upgrade the OS from “Centos 6” to “Centos 7”

1.Create Backup for the existing OS server.

In both cases “On primse or AWS”, we need to create backup for the OS before we upgrade it.

So if we working with **on premise** infrastructure we can create a backup for the hole OS “OS root file system + attached HDs” in separate storage.

And if we working on **AWS** as our infrastructure we need to create snapshot for hole EC2 before upgrading the OS and by default it’s store in S3 storage.

After you backup your data now you can use one of these two ways:

2.Upgrade your deployment directly, For CentOS, there are two steps in this process:

1.Run the 'Preupgrade Assistant' tool, This will try to find any problems that might come up when upgrading your CentOS.

Run the upgrade tool to actually complete the upgrade.

This is covered in more detail in the following Linux Academy link: CentOS 7: Upgrading from CentOS 6.x In Place

2.Instead of directly upgrading, you can create a new CentOS 7 deployment and then copy your data over the new OS.

I recommend option number 2 to create a new Vm or instance with the new version of CentOS7 and Copy the data and configuration files.

To save least Downtime we can use the Advantage of using LB here.

Using LB can help us more for forwarding the traffics between the OSs.
For example if we have 5 Web servers that managed through HA-proxy or varnish using mechanism like Round Robin to forward the traffic.

We can remove the OS IP in HA proxy configuration file, to disallow the traffic to pass on this OS.
After our update and check finish successfully, we can back the VM again to our LB configuration service.

Using AWS infrastructure can help us also through Autoscaling , LB and Rout53 service can handle everything for us.

2.Upgrade the Mysql Package from “5.5” to “5.7”

1.Create Backup for the existing DB server.

In Case we using **On premise** Infrastructure we can create backup of hole OS or in case we using Mysql with “All On One” option, we can create a dump from the Database through “mysqldump command” only and move it to our backup server.

In Case we using **AWS** service, I recommend using “Aurora” as a DB service that provided through AWS DB portal.
Also AWS provide Automatic backup option for DB service, and also we can create a manual snapshot for it.

2.Upgrade the Mysql Package.

1.In case we using 3 DB instance “1 Master” “2 Slaves”, so we can assign one of the slaves instances to be the master node, and start to upgrade mysql package of master node and test the performance of using it.

if it successfully work we can point it back and start to upgrade the Slaves with the same idea.

The steps of upgrading very simple. just stop the mysql service, add the repo of new version, start to upgrade the package and start the service again.

Also we can create a new OS, start install the new version of mysql DB and import latest backup of instance that working.

We should do that in Maintenance Window to Guaranteed that we not lost records

2.In case we using AWS DB service, we can choose the desired version through AWS console and AWS will take care about the process.

3.Using “**Percona**” tool to upgrade the mysql package
https://www.percona.com/doc/percona-server/5.7/upgrading_guide_56_57.html#rpm-based-distributions

3.Upgrade the PHP Package from “5.5” to “7”

Like upgrading Mysql Packages, we need to create backup for the hole OS that have PHP packages, and it’s will depend on our infrastructure.

We can start using “Remi” repo or “Webtatic” Repo to install the latest version for PHP, For me i recommend to use the Webtatic repo, it’s more stable.

We need to specify the PHP modules that using in PHP version and check how it upgraded in the last version.

Note: in some cases PHP modules change in upgrading for example in PHP v5 the php mysql module called php56w-mysql and in PHP v7 called php71w-mysqlnd so be carefull.