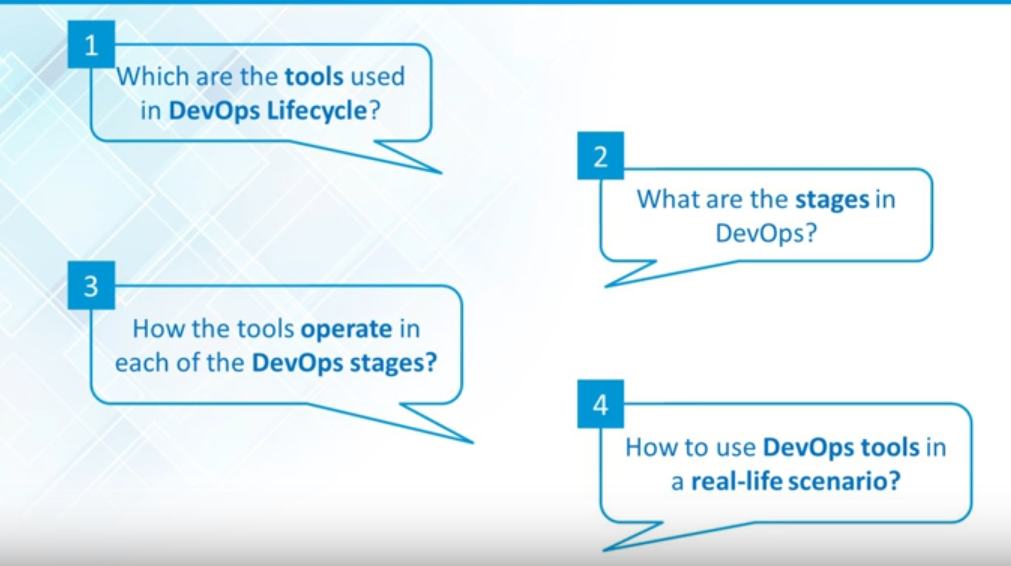
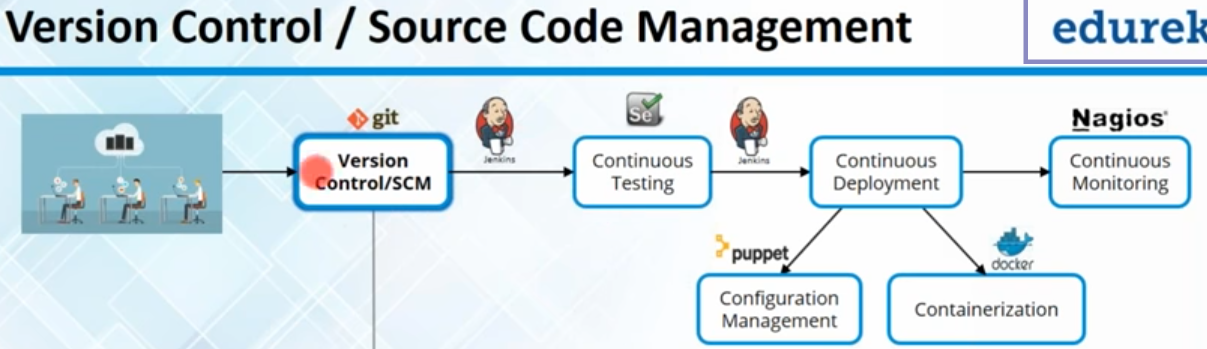
Automation using DevOps Tools

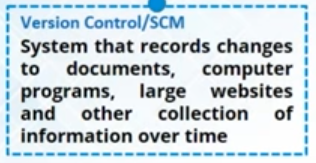


1. Stages in Devops
2. Which Tools
3. How tools used
4. Usecase

**Devops stages in real world**

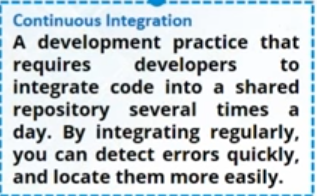


Stage-1: Developers commits code🡪Goes to Version control Stage (1st stage in Devops).



AT this stage, we can manage Code written by Developers.

Stage-2: (Continuous Integration)

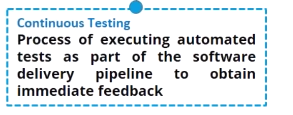


Rather than waiting for all developers to commit their code->then build & testing it,

In CI approach: When any developer commits code, that code is merged with main branch🡪 then we can use Jenkins to create build =>Test in virtual environment.

* It saves lot of time in moving Code from Dev to Test Environment.
* We need not wait for all developers to commit
* Even though, we are able to do CI, it is of no use if we are not testing continuously.

**Stage-3:** Continuous Testing



In this stage, we can automate tests by running test scripts using Selenium, and run in virtualized environment.

If test fails, we stop here and send issues to developers.

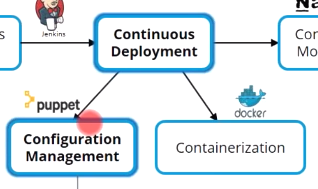
If passes, we move ahead and deploy it.

|  |
| --- |
| **Release** is to make the product/build available for testing before deployment, usually sent from the development team.  In-between release and deployment is usually the **testing and verification** process. After that cycle is done, we move on to...  **Deployment** is to send the build out (usually Gold/Master) for either updates or installation, usually handled by whomever is in charge of the deployment process. |

**Stage-4: (Continuous Deployment)**

As a Network Specialist, typical tasks I'd expect involved in "deploying" a new server would be:

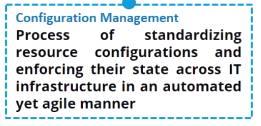
1. Buy server
2. Install OS
3. Install applications
4. Configure network settings (IP address, default gateway, etc)
5. Patch the server into the LAN switch
6. Configure appropriate firewall rules to allow required access to the server.
7. Add server into monitoring and management platforms



In Continuous Deployment, we have 2 parts:

1. Configuration Management
2. Containerization

**Configuration Management:**



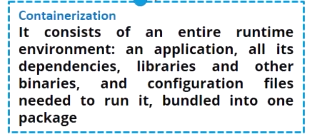
Consider there are 5 servers Up & Running.

They have same environment libraries, tools on them.

In older times, we used shell script to do the job. Now, if I need 100 servers with exact same configuration.., using shell script would be obviously complex.

We need some tool to do job for me. Puppet is more popular tool for Configuration Management.

**Containerization**



In this stage, we can create test environment, where we can run your application and test it.

This stage is mostly used for test purpose. It gives virtualized environment where I can run my selenium scripts to test my application.

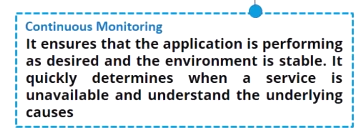
90% of organizations are using Containerization in testing stage.

10%-> production. In late 2015, docker-1.9 came, which is stable to run in production

For ex: if we need to run few applications on Prod for shorter duration.., U can easily create a container on Prod...& run those applications on container.

When we don’t need that application anymore, we can destroy that container..

**Stage-5: (Continuous Monitoring)**



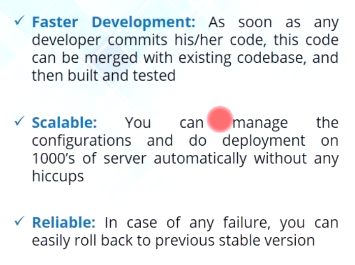
Now that we have deployed our application, we need to monitor it.

If application is performing as desired and environment is also stable.

By monitoring application, we can quickly determine when service is unavailable, and understand underlying causes…Then we can report all issues to develoepers

Devopers Code-> Push to GIT(Version ctrl/manage code->Build(**Jenkins**)->automated Testcases(Selenium) **using Jenkins**->Deploy on Prod, where we use:

Configuration Mgmt: puppet🡪 deploy on 1000 of servers 🡪Monitor



Advantages,using Devops

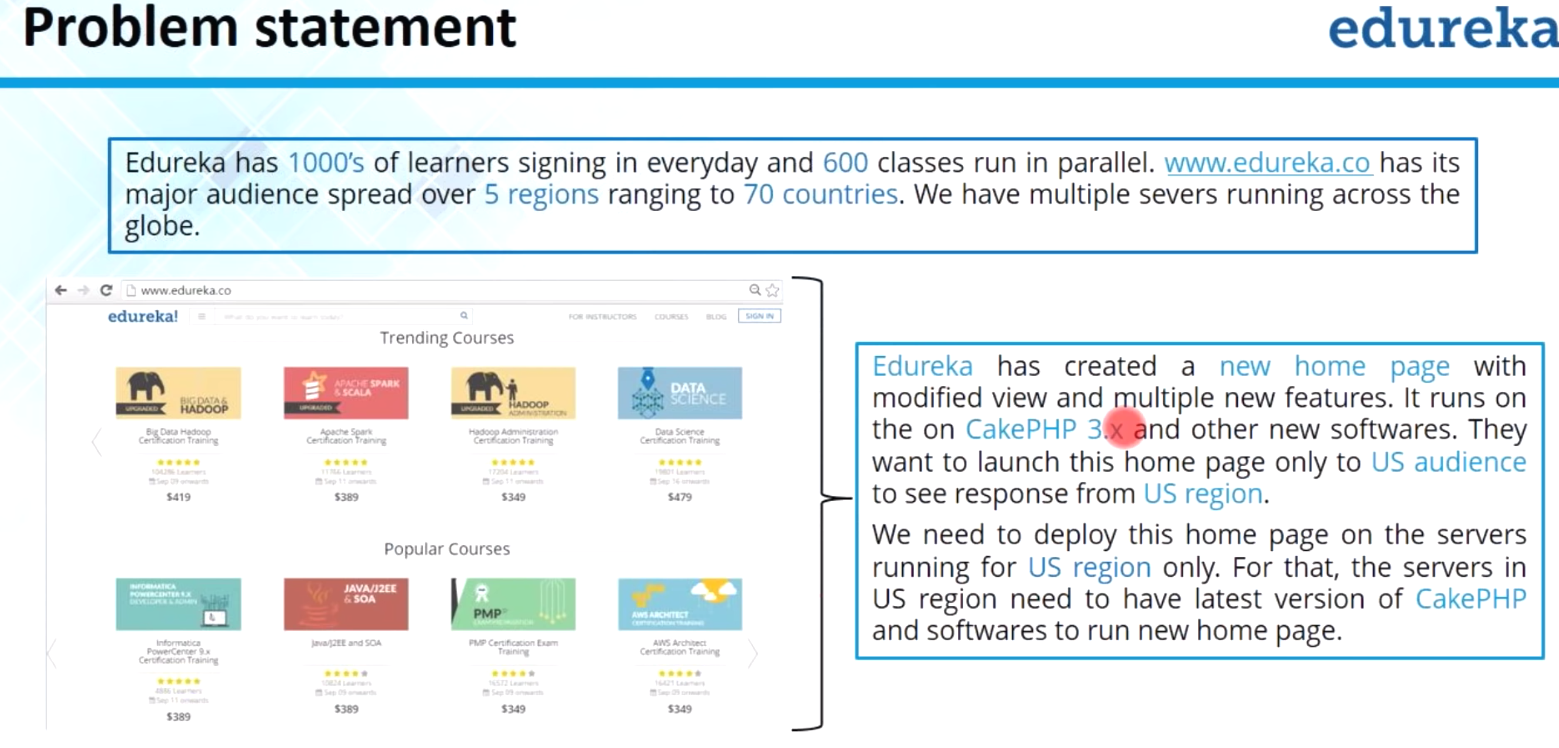
1. We don’t need to wait for all developers to commit code.
2. Using puppet we can deploy 1000 of servers.
3. Once we deploy your application on Prod, if something goes wrong.., can rollback to previous stable version.



Think about scenario, where we have launched new homepage with personalized features...

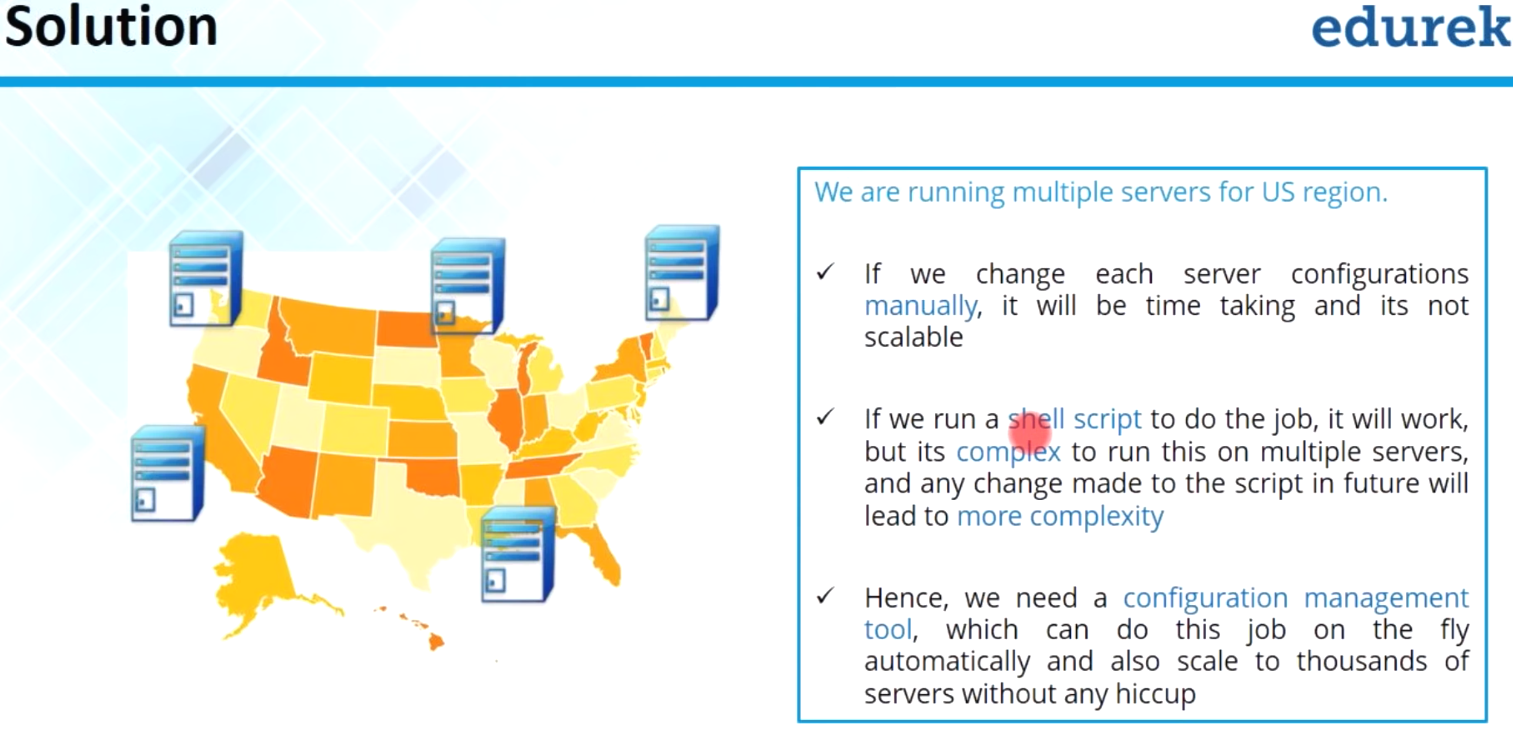
Before making available to all users, we want to roll to selected users.. To check everything is working fine, before wider deployment.

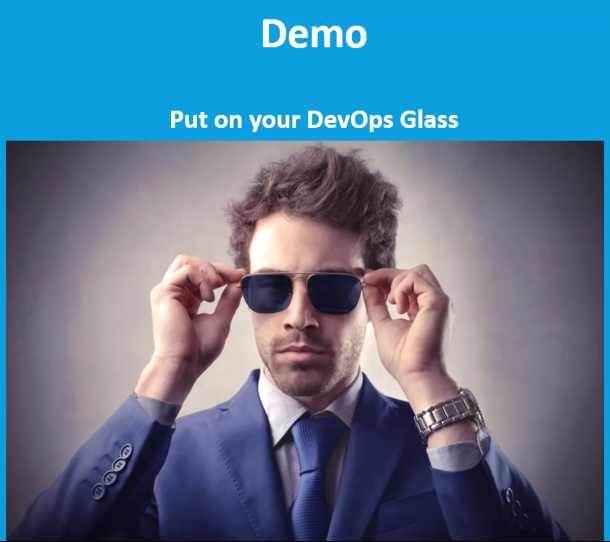
This approach is called dark-launching. This approach is coined by Facebook.



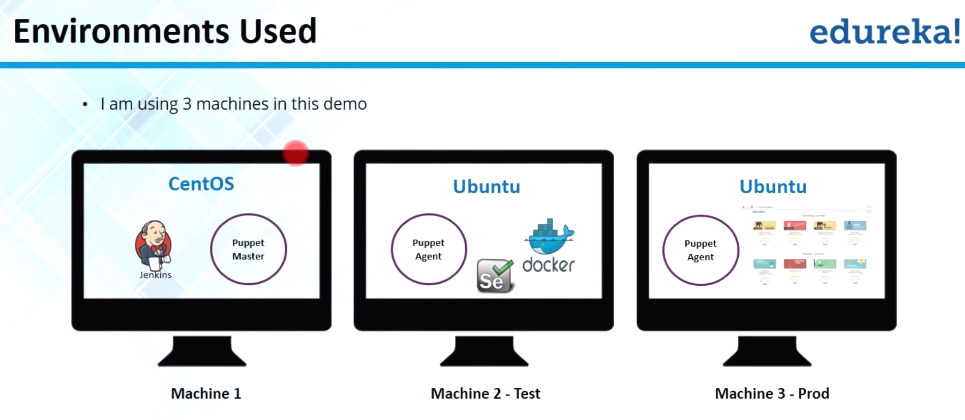
Let say we are using for US audience only. So we deploy this on US servers..

Another difference is we want to migrate to Cake PHP 3.







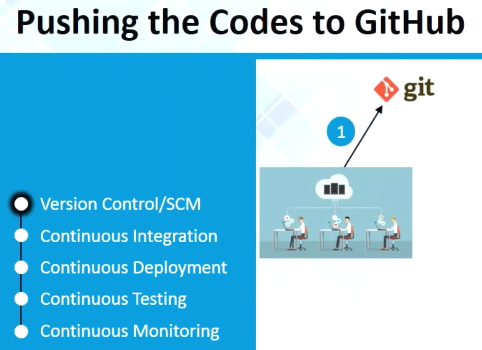


1st: Centos-> Jenkins, Puppet master

2nd Ubuntu(TEST):->Test environment using Docker, Run application in docker container,

Puppet agent is also running on this machine. will run a testcase for login as well by running Selenium scripts through Jenkins.

3nd Ubuntu(PROD)->Puppet agent running, This is where, I’m going to deploy my application finally.



My project is already available on github, whatever developers pushed to github

