**Objectives**

* Demonstrate the steps involved to deploy an application in a remote server
  + identify software required for remote, identify steps to deploy an application

* Explain the need and benefit of Docker
  + open source; create, deploy and run applications in containers; package application; virtual machine; image; container
    - Reference: https://opensource.com/resources/what-docker

**Understand deployment of application in a remote host - PART 1**   
  
**Scenario**

* The applications spring-learn needs to be deployed in a remote windows server
* This servers can be accessed through Remote Desktop
* Except Windows Server operating system no other software is installed in this server

**Objective**

* Identify the list of software to be installed in the server for deploying spring-learn application [HINT: development tools should not be installed in the server (example: Eclipse)]

**Group Formation Guidelines for SME**  
Find the answer for the question in the above objective as a group activity.

* A group size should be four. If not possible to equally divide, few groups can have size of three.
* To randomize the group members, use the online tool available at  <https://www.aschool.us/random/random-pair.php>
* Enter “Max group size” as 4
* In the text area after “balance”, copy and paste learners list of the classroom. Each line should have one name
* Click “Submit” which will display the groups and the list of people in each group
* Project the screen and show the groups

**Expectation from each Group (Duration: 15 minutes)**

* Identify a leader in your group
* Identify a name for your group
* Each team member to share their point of view on the list of software required
* After discussion and consensus arrive at the final list of software required
* Write the final list of software in a paper under the below category:
  + For deploying spring-learn

**Finalizing the software list**

* The group leader has to convey the team name and the software list to the SME
* SME has to write the group name and the software list on the board under the respective team name
* Once all the teams had provided the details, the SME has to brainstorm and finalize the list of software required.

**Understand deployment of application in a remote server - PART 2**   
  
**Activity**

* List out the components for deploying spring-learn application with eureka discovery and zuul gateway
* For each component list out the steps required for deployment
* With the assumption that your PC has only the list of software identified in the previous activity
* The list of steps should only use the software identified and should not use any development tools
* Continue working as same groups formed in the previous activity
* The group lead of each team has to collect inputs from each team member, debate and arrive at the final list

**Expected Output in a white sheet**

* Component 1
  + Deployment step 1
  + Deployment step 2
* Component 2
  + Deployment step 1
  + Deployment step 2
  + Deployment step 3
* Component 3
  + Deployment step 1

**Activity Finalization**

* SME to collect inputs from each group
* The entire classroom has to collectively finalize the list of activities and write the list of activities on the board

**Deploy application in Desktop PC**   
  
**Scenario and Tasks**

* Assume your PC as remote server
* Refer activities from previous exercise and perform the deployment steps in your PC. (NOTE: Ensure that development tools are not used during deployment)
* Based on those steps ensure that spring-learn application works end to end
* Copy spring-learn projects to a new folder C:\Users\[EMP\_ID]\build. The perform the build and deployment steps for achieving the above objective

**Optimizing our deployment steps**   
  
Now we know the steps for deployment in a remote server, let us ponder over the below questions. 

* How much time is required to deploy our application in the remote server?

* Is there a possibility to have human error during deployment?

* Is it possible to automate deployment steps? (Find available tools by searching internet with keywords "deployment automation tools")

* In the remote server, JRE 11 is already installed and is used by an application. Our application is tested in JRE 8 and not tested in JRE 11. How do we address this issue? (Containerization addresses this issue, learn and find out various tools available by using the search keyword "container deployment tools")

**What is Docker?**   
  
Let us do a simple exercise before trying to understand Docker and it's benefits.

* Execute the following commands. The explanation for these statements will be provided later:

docker image ls

docker container ls -a

* Open command prompt and execute the following command.

docker run -it debian

* This command downloads linux debian operating system from docker hub (<https://hub.docker.com/>) and runs linux operating system within the local docker server.
* A linux operating system command prompt opens up and should look something similar to this:

root@86bdf44d1bd7:/#

* Let us execute few linux commands to find out how it works.
  + Print current directory

pwd

* List all files and folders in the current directory:

ls

* List system information

uname -a

* Exit from debian linux

exit

**Key take away points**

* One can understand how quickly we have launched an operating system
* The "docker run" command performs the following tasks:
  + Downloads debian ***image*** from https://hub.docker.com
  + Runs the debian image in a ***container***
  + The -it option provides an interactive command prompt
* An image in docker contains the binary version of a software application
* A container is created for creating an working environment based on the image
* To view the list of images and containers execute the following commands, which will display the image and container details for debian
* In a similar fashion we can roll out JDK, MySQL, etc. on top of this linux platform and run them in containers.

**What Docker is?**

* Docker is open source
* Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.
* Docker allows a developer to package an application
* By testing in Docker, a developer can be rest assured that the code will work as expected in a linux platform without having a need to have an actual platform.
* Docker is like a virtual machine
* Benefits both developers and system administrators
* This forms part of DevOps toolchains.