SELENIUM GRID & SauceLabs

1. **What is Selenium Grid?**
   - Gives you the ability to run your automated tests in different browsers(and their different versions) and platforms (basically Operating systems and their versions. Window, Linux, Mac) VISGRID
   - This tool is useful if you have lots of Tests (over 500)
   - Instead of doing

```
WebDriver driver=new Chromedriver()
```

   - Do

```
WebDriver driver=new RemoteWebDriver(url,capabilities) //Contains 2 parameters in constructors
```

2. **When do you use Selenium Grid?**
   - Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution

3. **How does Selenium Grid work?**
   - Grid is a set up that consists of Hub and nodes
   - Hub is a central machine that all nodes connect to
     - Has IP address and port number, Then you link the Hub to Nodes
   - There is a main machine which is called a Hub and multiple nodes (the machines that actually run your tests)
   - The order of execution is;
     - Your code > remote driver > Selenium Hub > Selenium nodes (might be multiple ones)
       - You asked your Selenium Hub to run your tests
         - Then selenium hub will find a node that is linked to the hub and run your test from there
       - You can have as many nodes as you want but only one hub

4. **How do you set up your selenium grid?**
   - Hub is a different server machine and each node is a separate server machine
   - Our hub and nodes will be set up in Amazon AWS Ec2 machines (ideally)
   - You can ask your hub to run tests and then the hub will find a node and run your test from there
   - We can also have hub and multiple nodes in same machine or server
   - Each node registers to the HUB with a certain configuration and HUB is aware of the browser available on the node
   - When a request comes to the HUB for a specific browser(with Desired capabilities object), the HUB, if found a match for the requested browser, redirects the call to *that* particular GRID Node and then a session is established bidirectionally and execution starts

5. **Where are the cross browser/platform running?**
   - It is running in Amazon AWS machine where Jenkins is installed. But normally, company jenkins is used by developers' team, devops, deployment team, and QA automation team
     - It is not recommended there
   - Ideally is Jenkins > GitHub > Maven > Runner class>Selenium Hub and run in one of the nodes that is configured in different server
     - In your hooks class, instead of having you webDriver point to a local driver, change it to a RemoteDriver(); and have it point to the cloud machine that has the hub

6. **What are the challenges in multi browser testing?**
    - Something is not clicking
    - Not visible
    - Some items look different in one browser than another browser

7. **What browsers are you testing?**
    - Chrome - Firefox - IE/Edge - Safari - Opera

8. **What is a hub in Selenium Grid?**
    - A hub is server or a central point that controls the test executions on different machines

9. **What is node in Selenium Grid?**
    - Node is the machine, which is attached to the hub, There can be multiple nodes in Selenium Grid.

10. **How do you automate multi browser testing?**
    - Change the browser to something else in my properties file in my framework
        - When I want to run my tests in different browser
        - Ex; "Browser=chrome" to "Internet Explorer"
        - This method works if your tests are less than 500 tests
    - In my framework, I implemented Selenium Grid and I can run tests in different cloud machines using different browsers

11. **What to do with failed tests?**
    - Look at the automation execution report
    - Find out the reason of failure
    - Try to do steps manually,
        - If manual is passing then automation issue → so you fix, re-run and see if its passing
        - If its application issue
            - Create a defect
                - While the defect is being fixed, i am testing manually using Ad-hoc test
        - If the defect is not a showstopper then you run other tests and automate
            - If it is, you have to wait, can't do any further testing
        - When rerunning, I use @ReRun tag to test only the test you want to rerun

12. **What is Ad-Hoc test?**
    - Performed without proper planning and documentation
    - Defects found using this method are hard to replicate since there no test cases aligned for those scenarios
    - Performed after formal test execution

13. **SauceLabs - cloud Grid service. Access to multiple Platforms with multiple browsers**
    - Provides cloud machine so you can run a lot of
    - Does parallel testing well

14. **How do you report using SauceLabs?**
    - I link to my JIRA server

15. **How to test with SauceLabs?**
    - Usually, we do "WebDriver driver=new FireFoxDriver();
    - Now we going to do;

```
DesiredCapabilities caps=DesiredCapabilities.firefox(); caps.setCapability("platform","Windows 7");
caps.setCapability("version","38.0");
WebDriver driver=new RemoteWebDriver(new
                  URL("http://YOUR_USERNAME:YOUR_ACCESS_@ondemand.saucelabds.com:80/wd/hub",ca ps)
```

16. **Steps to run your tests in saucelabs/Any ready selenium Grid**
    - Create desiredCapabilities object and specify with type of OS, browser you want your tests to run with selenium Grid.

```
DesiredCapabilities caps=DesiredCapabilities.firefox(); caps.setCapability("platform","Windows 7");
caps.setCapability("version","38.0");
```

    - Create RemoteWebDriver with HUB url:

```
WebDriver driver=new RemoteWebDriver(new URL(URLOFHub),caps);
```

    - Create sauceLabsDemo class

```
// In @BeforeTest
// DesiredCapabilities(comes from Selenium)
caps=DesiredCapabilities.firefox(); //(chooses which browser)
caps.setCapability("platform","Windows 7");
caps.setCapability("version","38.0");
```

    - String URL="http://YOUR_USERNAME:YOUR_ACCESS_@ondemand.saucelabds.com:80/wd/hub"
        - This is Selenium Hub address
        - **URI**= unified resource identifier
        - **URL**=unified resource locator

17. **How do you run your tests in multiple threads parallelly?**
    - Thread is like one process or instance of application run
    - 4ways
        - 1. We can create multiple cukesrunner with different tags
            - Ex. cukerunner has tag "@Test"
            - Smokerunner has @smoke
            - Regression runner has @Regression
        - 2. Create testng xml and add those runner class under one test
            - Add all 3 runners in one xml
            - Then next to verbose=2(on top of xml file) add "parallel="classes" thread-count="10">
        - 3. Then make sure our driver opens a remote WebDriver that is point to hub
            - Add the code in driver class
        - 4. Run the testng xml by itself or using maven.

18. How do you set up Selenium Grid in AWS?

- (1) 2 cloud servers (Ec2)
  - 1 will be HUB
  - 1 will be node
- (2) In machine 1, download Selenium StandAloneServer
  - It is needed in order to run Remote Selenium WebDriver
    - A lot of configuration
    - Use command line
      - Set up HUB;
        java -jar selenium-server-standaline-3.5.3.jar - role hub http://localhost:444/grid/console
      - Set up node1
        java -jar selenium-server-standalone-3.5.3.jar -role node - hub http://locahost:4444/grid/register
      - Node2
        java -jar selenium-server-standaline-3.5.3.jar -role node -hub http://localhost:4444/gird/register -port 7777
- (3) You can also use Visgrid
  - Download in machine 1
  - Open jar file
    - Set max session to 10
    - Start hub
    - Port 4444 (where your hub is. You can change it but remember it)
  - Click Create node
    - Choose a browser for the node
    - Type in number of instances
    - Click Add
      - Open browser in ec2 machine
      - Type: localhost:4444 (this is seleniumGrid hub) (Now we just powered up our HUB)
      - Click on console
      - Refresh the page
      - Now it Show nodes (all in the same machine)
      - Add another node(now it will show 2 nodes)
  - Now go to another machine and create nodes from there and link it to HUB
  - Before going to machine 2
    - Go to aws console
    - Go to the instance that hold your hub
      - Click on security groups = launch wizard
      - Go to inbound
        - Add the port number of hub (which is 4444) and save it
  - Go to machine 2
    - Download same jar file of Visgrid you used for 1st machine
    - Download jdk (in order to open the file)
    - Open jar> start hub>create node>click Override HUB(we don't need this hub since we have another one on the first machine)>add machine 1 ip address and port number: 4444>add

19. Linux Commands (case-sensitive)

- reboot → reboots system
- man → gives you instruction of the command - Ex: "man reboot"
- mikdir → Creates directory(folder)
- cd → Change directory
- Ls → List directory content
- pwd → Print name of the current working directory. It gives you the exact location; Ex: /home/Andy/Desktop
- ll → Long list format
- ls-la → Prints files and hidden file
- clear → Clear screen
- cd.. → Goes to the parent file (not the root file)
- cd/ → Goes to the parent root file
- cd~ → Goes to the home of the user file
- grep → Prints a line matching a pattern
- df-h → Prints the disk space usage
- top → Displays linux tasks (like task manager)

- **How to create an account**
  - User → useradd Andy
  - Group → groupadd Cybertek

- **Adding a user into group**
  - useradd -G Cybertek Andy
  - id Andy → prints details for this individual (shows it Andy has cybertek)

- **Configuration/changing the network**
  - vi /etc/sysconfig/network
  - crontab → Sets a timer for your file to run (build schedule likejenkins)

- **Setting permissions**
  - chmod → Change file mode bits
    - Order is owner, group, others
      - If the file is folder, d is in the front
      - If not a folder there is no d
  - chmod 777 → Gives access to owner,group, and others; Very dangerous; DON'T USE
    - r- read
    - W-write
    - X-execute
    - rwxrwxrwx (777)
  - chmod 644 → Give access to owner (read and write),group(read only),others(read only)
    - -rw-r--r--
    - Default access and Standard
- How to find an file using grep (if you know the name of file)
  - grep 'name of test failed file' /home/Andy/Test1/TestScenario (location) > /home/Alex/AutomationFile
    - Now the file is inside this location; /home/Alex/AutomationFile
  - Find any file that has ' name'
  - grep 'fail scenario' *