# DevOps Engineer/Site Reliability Engineer

**HR**: Please send this questionnaire to the candidate a week before the interviewer and interviewee can discuss the solutions in a conference call

**General**: The solutions to the below questions will be discussed on a conference call through a desktop sharing session. The interviewee will display the execution of code on his/her machine during the conference call.

**Interviewer**:  The interviewer may choose to ask other similar questions during the interview process

**Interviewee**:  You may search online and find solutions to the below. Answer as many questions as you can in the time provided. You may use the programming language of your choice. For each of the questions below, create a repository (preferably git) and share the solutions with us at least a day before actual discussion. You will have to share your desktop while explaining the solution. You are expected to set up, configure and have the environment (infrastructure, runtime and libraries etc.) ready to solve the below questions on your personal desktop/laptop/cloud.

---

1. Write a program to reverse a string “abcdef” --> “fedcba”  (preferably NOT using inbuilt functions that come along with the libraries. We want to understand how you would think algorithm wise)
2. Write a program that can parse an integer array and verify that is of social security number format. SSN number format is [3 digits – 2 digits – 4 digits] Ex: 123-45-5678
3. Write shell script that does the following in sequence (on a *nix box)
    a. checkout code from github into folder /home/testuser/mycode
    b. Assume there is a config.json file in the source code with a JSON structure {user: , api_key: ‘1234’ , conn_string: , ip_address: }. Also assume that the values for the keys in this JSON come from variables $user,$api_key and so on. Assign these variables to the values of corresponding JSON keys ONLY if the value is empty string. So the config.json should now be populated with values [This step needn’t necessarily be in shell script, assume you have /usr/bin/ruby OR /usr/bin/python OR /usr/bin/java OR /usr/bin/node in PATH]
    c. Change the ownership for all files on /home/testuser/mycode to username ‘testuser’
    d. Now archive this folder (.tar.gz), checksum and scp the archive to another machine that has dns ‘remote.test.com” (in location /home/testuser/remotecode) . Assume that ‘testuser’ is part of sshlogin group on remote.test.com
    e. Stop the service with name ‘node’
    f. Move the archive to /user/node/data and unarchive the contents
    g. Start the service with name ‘node’
    h. Check that the end point ‘http://remote.test.com/status’ returns code 200
    i. Read about selenium grid online

4. Read about Selenium GRID online
    a. The GRID is a java jar that has an embedded jetty server that can be started into two roles aka. Hub and node.Bring up a GRID on your localhost with hub running on 4444 and two nodes each running on 5556 and 5557. Accessing the hub on http://localhost:4444/grid/console will display a dashboard with all available browsers as configured on nodes. Hence launch the nodes with 5 instances of each browser type i.e. chrome, firefox and ie (Any versions of the browsers are fine). If you would have to launch the above jvm’s as a service on windows & linux, how would you go about it?
    b. Note: Bonus points (not necessary though) if you can write a simple client script (java, ruby, python, php ..any language) that can launch a browser using this GRID and launches a chrome browser and hits the website http://www.inmar.com

5. Assume you have multiple log files within a folder. The content in the log file is being written by an application and is space delimited. The fields are [UTC_TIME CLASS LOG_LEVEL  DESCRIPTION]. An example is

```
Apr 30, 2015 8:27:17 PM org.apache.catalina.core.ApplicationContext log
INFO: ContextListener: contextDestroyed()
```

Write a shell script that can find all log entries in this folder that are of LOG_LEVEL=severe and in between “Apr 27, 2015 8:00:00 PM” to “Apr 30, 2015 8:00:00 PM”. Redirect the output of this script to a file “test.txt”. The file “test.txt” should contains 3 fields viz. name of log file, # of log entries and the actual log entry contents [Don’t worry about the formatting]

## Version Control

- Explain version control strategies that you worked with, preferably Subversion, Git , Mercurial etc. Demonstrate on command line, how you would achieve branching, tagging, resolving merge conflicts. You may create a hypothetical repository for demonstration.

## Configuration Tools

- Demonstrate with execution,  the most complex software configuration task you have automated so far. Use any of the configuration/deployment tools like Ansible, Chef, Salt, Puppet etc.
- Using any of your favorite tools, demonstrate the concepts of Continuous Integration, Continuous Deployment and/or Continuous Test Automation (e.g. you can stand up a Jenkins server and show us how you create a job. Of course, you are free to use other tools, jenkins happens to be one)

## AWS

- Set up a bastion host (jumpbox) in AWS , so that users do NOT directly login to production server instances, but instead are routed through the bastion host for security reasons.
- Demonstrate with execution how you would set up an ELB and its purpose (What is the difference between application load balancer and network load balancer that AWS provides?)

## Application Monitoring

- Demonstrate with execution, the most complex application monitoring solution that you have used and show us the metrics that you would look for during troubleshooting. If you have used any paid/proprietary solutions like NewRelic, Datadog etc. and can't show us the dashboards, at least walk us through a scenario of what metrics did you look at and how did it help identify the root cause.

## Container(s)

- Walk us through your container environment setup (docker-engine preferably) and how would you use containers to achieve speed ? If you have used any container orchestration solutions like Swarm, Kubernetes, Mesos, Rancher etc., please demonstrate by showing us a setup and the features that you feel would be beneficial for administering and maintaining such an ecosystem.