

CROSS | OVER

Congratulations on making it to this stage of the evaluation. You are obviously very talented as very few people make it to this stage. As we've stated earlier, the companies we represent receive 1000s of resumes for any given role and it is through these difficult assignments where you can differentiate yourself and be noticed. After completion of this final 'real scenario' assignment - there will be a quick technical interview on your delivery then you are ready to be hired.

The project is scoped to be simple and reasonable in size to enable you to demonstrate your enterprise - class skills. Though this is a fictitious example, this scenario is very similar to what you may encounter in your job.

Instructions

- Try to complete as much as possible within the given time frame. If you need more time, please ask for an extension. You must complete full-functionality of the application with industry-level coding style/commenting. Unfinished assignments will not be considered.
- Please note that you are expected to work on the assignment independently. Discussing assignment details with colleagues or any indication of outside help will be considered cheating.
- Please do not expect too much hand-holding as this is an evaluation assignment.
- Read the complete assignment before you start. Understand clearly what is required so that your work will be appropriate and easier.

Overall Objective

Create the architecture and design of a network based machine statistics collection system. Implement the system's applications.

Prerequisites

The following prerequisites should be respected.

1. You can code in python using any IDE you prefer.
2. Use any SQL database e.g. mySQL
3. Do not use any proprietary technologies or tools that are not available for free or for trial.

Functional Requirements

The system allows collection of machine statistics in an intranet environment. The system implements the following specifications.

1. Client Script

1. This script will be uploaded and executed to 100s of machines in the intranet. These machines are meant to be monitored for system level statistics like memory usage, CPU usage, total uptime and windows security event logs (in case of windows OS).
2. When executed, the client script collects the statistics and return them to the server script for cumulation.
3. The client script must encrypt the data before returning it to server.

2. Server Script

1. Installed on a single central machine in the same intranet.
2. Each client is configured in a server config xml file something like this

```
<client ip='127.0.0.1' port='22' username='user' password='password' mail="asa@asda.com">  
  <alert type="memory" limit="50%" />  
  <alert type="cpu" limit="20%" />  
</client>
```
3. When executed, the server script should connect to each client using ssh, upload the client script in a temp directory, execute it and get the response.
4. After receiving the response from the client script, server script should decode it and stores it into a relational database along with client ip. This collected data is consumed by another application, that is out of scope, but you may design the database table yourself.
5. The server based upon the "alert" configuration of each client sends a mail notification. The notification is sent to the client configured email address using SMTP. Use a simple text mail format with some detail about the alert. event logs must be sent via email every time without any condition.

Assume any functional details required to achieve the above requirements based on logic and your experience. But follow the KISS principle.

Other Technical and Non-functional Requirements

The following list of technical specifications should be adhered to

1. Apply validations and constraints wherever necessary to create a stable system.

2. Assume missing/unclear requirements to fill in the gaps in the specifications.
3. You can use paramiko for ssh communication.
4. You can use win32api for statistics.
5. You must use SMTP to send emails. You can use smtplib for this purpose.
6. You can use pycrypto for encryption purposes.
7. You should write unit tests for your application.

What we will evaluate

1. Efficacy of your submission: fundamentally how well your solution is able to achieve the assignment objective and solve the stated problem.
2. Code quality
 1. Code modularity
 2. Application organization.
 3. Code documentation - balancing between self documenting code and comments
 4. Unit and integration testing
 5. Exception handling where available and expected.
3. Design
 1. Clarity and completeness of the readme and design documents
 2. Fitness of solution to problem
4. Functional completeness
5. Scoring ratio matrix (out of 10), all of these are individually mandatory so don't skip any:
 1. Design quality = 2
 2. Code quality = 3
 3. Docs and demo quality = 2
 4. Specifications compliance = 3

What to deliver

Demonstration Video

Record the video demonstration of your work using a screencast tool like [screen-o-matic](#) (or any other tool you prefer) commenting on the execution of all components. Save the video to your local machine and include it with the delivery package.

Database script

Create manual steps and SQL script files to create the database (if required), its schema, stored procedures, or any seed data you have used for testing. Also do not forget to mention the steps to run the scripts in the readme.

Readme Document

Create a txt file with the following information

1. Instructions to install and configure prerequisites or dependencies, if any
2. Instructions to create and initialize the database (if required)
3. Assumptions you have made - it is good to explain your thought process and the assumptions you have made
4. Requirements that you have not covered in your submission, if any
5. Issues you have faced while completing the assignment, if any
6. Constructive feedback for improving the assignment

Design Document

Create a design document containing the following

1. List of all the libraries / packages you have used in your project.
2. High level requirement analysis
3. Architecture diagrams describing the composition and working of the system, explaining the component interaction and process, control and data flows.
4. Explain the breakdown of the system into components with technical implementation details of each component along with the design patterns involved and with reasons that justify your choices.
5. Use both visual elements (diagrams) and text descriptions to maximize the amount of information conveyed while keeping the document as compact as possible

Source Code

You should deliver all the implemented source code.

What to submit

Please read this section carefully.

Failing to follow these directions will disqualify you from consideration.

Create and submit an archive named `<your_name>_SR_Python_SysInfo.zip` containing the following:

- `<your_name>_SR_Python_SysInfo.zip`
- `<your_name>_SR_Python_SysInfo.zip \Readme.txt`
- `<your_name>_SR_Python_SysInfo.zip \Design.doc` (or pdf etc)
- `<your_name>_SR_Python_SysInfo.zip \Demo\` < this folder contains the screencast video
- `<your_name>_SR_Python_SysInfo.zip \Source\` < this folder contains the complete source code

ATTENTION! YOUR APPLICATION WILL BE REJECTED IF IT:

- Failed to execute properly, without errors
- Does not contain unit tests
- Unit tests are failing