QA Engineering homework/project

Homework for potential QA engineer.

Setup

1. **Hosts**
    a. **Agent** - 1 host(Tagged ‘Agent’) where agent software (provided by Cribl) will be installed.
    b. **Splitter** - 1 host(Tagged ‘Splitter’) where same above software is configured to split data stream will be installed. This will receive data from Agent host.
    c. **Target** - 2 hosts(Tagged ‘Target-1’ & ‘Target-2’) running above software where data from ‘Splitter’ will arrive and data will be kept in file system.
2. **Applications**
    a. **Agent** - An application on Agent host will read from a file (specified via configuration). This agent will send the data from the file to Splitter (also specified via configuration). Imagine that this application already exists. Details will be provided in ‘Objective’ section.
    b. **Splitter** - This application will send the data to final destination on 2 Target hosts by splitting data in random manner and send to 2 hosts, Target-1 and Target-2
    c. **Target** - Application on target machines, will receive data from Splitter host and write to a configured file.

See qa-project-setup.jpg for diagram of setup.

Objectives

Automate the following tasks using language of your choice:

1. Download a nodejs application that works in **Agent**, **Splitter** and **Target** mode. Download the application and install them on 1 Agent, 1 Filter and 2 Target hosts.
2. Install them somewhere on the each of the hosts.
3. Configure them as below:
    a. **Agent** host (sample configurations provided. Make sure to fill in appropriate host name) -
        i. A JSON file ‘app.json’ with content {“mode” : “agent”} to configure application in Agent mode.
        ii. A JSON file ‘inputs.json’ with content `{“monitor” : "inputs/large_1M_events.log”}`in directory where Agent application was downloaded.
        iii. A JSON file ‘outputs.json’ with content `{“tcp” : { “host": "$SplitterHost", "port" : 9997”} }` in directory where Agent application was downloaded.
    b. **Splitter** host -
        i. A JSON file ‘app.json’ with content `{“mode” : “splitter”}` in the same directory as the application to configure application in Filter mode.
        ii. A JSON file similar to above ‘inputs.json’ with content `{“tcp” : “9997”}` in the same directory as the application.
        iii. A JSON file ‘outputs.json’ with content `{“tcp” : [{"host:"target_1", port:9997},{"host":"target_2", port:9997}]` in the same directory as the application.
    c. **Target** host -
        i. A JSON file ‘app.json’ with content `{“mode” : “target”}` to configure application in Target mode.
        ii. A JSON file ‘inputs.json’ with content `{“tcp” : “9997”}` in the same directory as the application.
        iii. A JSON file ‘outputs.json’ with content `{“file” : “events.log”}` in the same directory as the application.
4. Start the applications in the order : Targets, Filter, Agent by running `node app.js <conf_directory>`. It is important to start in this order, or the application may not work properly. You can pass “target”, “splitter” or “agent” as the directory. These directories already contain sample configuration files. Make sure to set correct host names in outputs.json.
5. The Automated integration test should verify:
    a. Count of Events in 2 Target host file ‘events.log’ is 1M. If count is wrong, automated test should declare the test failure.
    b. Should capture output from standard out of application
    c. Record artifacts from application.

Running app as agent/splitter/target

This app require **node** 12 or higher. You can run the app as:

- **node app.js <conf_dir>** where <conf_dir> contains app.json which determines the mode the application runs in.

Deliverables

- Script that automates the process. You should be able to explain to us why you decided this approach vs other approaches that you may have considered.
- Demo of the process in action.