Assignment for the technical interview

General information

Infrastructure for this exercise consists of the servers running in the AWS cloud. It emulates typical setup of the CD pipeline. The following servers are used:

- Git server - is used to keep Git repositories. See **git-01** below.
- CD server - is used to orchestrate delivery process. See **jenkins-01** below.
- Static server - is used to run application (i.e. run-time production server). See **static-01** below.

Expected developer workflow is the following:

1. Developer clones repo from **git-01** to the local machine.
2. Developer makes a change in the code base.
3. Developer commits changes to master and pushes commit(s) to *origin*.
4. Developer logs in to the Jenkins UI that runs on **jenkins-01** server and runs the pipeline.
5. Pipeline builds the code, creates distribution package(s) and deploys them to **static-01**.

Assignment

You are expected to use tasks described below to show your problem solving skills as well as knowledge of technologies used in Continuous Delivery (CD) domain.

Provided solution shall be error free when used correctly. How everything else is handled is subject to your creativity and ambition.

Our JavaScript application (*netent-slot*) stored in the */srv/git/slot.git* repository on the *git-01* server used to be built and packaged using *npm* tool and some scripts in the *./scripts* directory. Created packages were deployed to the run-time server *static-01* using Jenkins pipeline on the *jenkins-01* server. After a major incident this environment was destroyed, but we managed to restore parts of it. Using information provided below, please, help us to recover remaining bits and pieces of the environment, so developers could start using it again.

Environment consists of the following servers running in the Amazon cloud:

	Public IP(*)	**Private IP(**)**	**Description**
git-01	`<provided separately>`	`<provided separately>`	RHEL7. Central Git repository. Accessible via SSH. Access URL is ssh://jenkins_job@<git-01>/srv/git/<repo.git>
jenkins-01	`<provided separately>`	`<provided separately>`	RHEL7. CD server. Runs Jenkins, available on port 8080.
static-01	`<provided separately>`	`<provided separately>`	RHEL7. Run-time server. Runs Apache, available on port 80.

(*) - public IPs for all servers in the environment are provided separately.

(**) - private IPs for all servers in the environment are provided separately.

Hint
To login to Jenkins master use `admin:<password provided separately>` Password is also available on **jenkins-01** `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

All the servers listed above have ssh access. It is possible to ssh into each of them (use public IPs) using *ec2-user* and the private key `cd-games-assignment-01.pem`:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAr4mkm
/csLgIJA6Q7bJTvf+Wsmsx8C0JyjA7DqvKeH5SfpuMeYf9fsfIZHDQD
vW7I1rAXP9Z000IA/AjPz/YuI5VmpaypfCvMlj//EPB5TlO3QIu2TGfPe3cNAnyWqxqIE2frr
//N
Zo1zZ7QzlqTjqUyV5LtpqAI3ghHF1PsNxHy+TDkmWiJOFF6fYOPMpQr/eBuispoRyN+XM
/CTATOG
vbjMQ82u4HbZBZlp/8iSwN+dihmI/Dfnm5hCtKVt+j
/x3Pn7AlSdirTKccIV7zjbFFWokmV0K6j6
qW9/oB7hAgth3GSvGuKp48B4tJG
/nAI+qzh3x3iCCz0FYPWH5ErUfwIDAQABAoIBAGTnwakqP8o+
v0HJoaM8RiSGOV9O1qt+Sex6
/D2DdCQBQnkjq653p2HaIlcVC2VUjrziwqqes89Q7NH0mskX4dPw
qO3pwA+x+cwRA6WQJptBAmFnLqbjMeDuQZtBVTcMbFaDcXddm4wtGb7uVo1tQbVrYer3VLgQdio
d
31Qb8QEWIyTCTHjnDxt2VIGo8hPYYgwMly4iPHqta3WOLiPd2F85gMpkbpoxDbtusZXcanuAwhz
6
NoMXAkD20WOK5qaLJ1Tan39UeJqZULo6Rjpo1u3957Q4EfAT9+zZ2nimLCQO6VowmP64G97joGo
0
jBywW48XeVIlWRgitvp4vqiby2kCgYEA6xQeomZbVFRiY4jAstJYYezmKcjIIarBOQAPNpb8MZh
G
fGjAbkN6EEI4BaTXefP
/w7nOLgZYlUH6yNUcw+MtDeXcPwvo29DRjNF21KjH2c4o5A6GoYraIJB1
ZzArvZPW0MY1SCflcn7okzZ1KRz4D8mNXu5UeUynGEG4NAmzphUCgYEAvyj7IhI1ucFAx0HQmAp
q
i4UIzh1rvu1LgiLvbC4CfZdumgHDX61cFRCEmvAxspesF
/AwefxdeXNUALCYu+lGUXxB0l+1MbZl
OeRO693xTFtSscJa2Y7xmmm/Ybis+Kw7jmwVyhIAk3hxx
/mTeETh26f9TdO74zJydfHJnbI5KUMC
gYEA2b0+DtBeIcMzitwdGMVKhhGmsuc9sCl4ZbKPZNH+8HUKPKvOKZOomSystVWVHGmL6p
/MwMay
ADYn8yOqTCq9DZVuJpuCaTCHGDbgKWPYEP8q0uo9e52gIu+LPzPYH
/Wy0Jf2vEnv9dhk1g5oANfL
UNwwbNC2P9BnKUmeM90UoG0CgYARswWtneoDjPUmfd2GdlBvIp4yF19GikoXfoyUg54Patn+wHM
K
hM6l1KWygtLXKlvwy3hxkwBEon0Ao1lA7NRp7wtZIytvsuekeJMmG6Kf40TA2LH4utd76Jajwam
i
TLAm+7L4xrnu7ZP7ZAxXcmlHIUK+1f
/rhVw7t2tngnzwaQKBgARyWxl2Bmf1DGHqj6EWQOWbLuWr
9gI9V0wRQtGcXOU9TXjIjABfKvprPw++8U7CDxcHzdP8SJkYkaxftVXJVtPfoVzasO9o3ONmrZC
f
RQji1PeNomV3IIUgG1CV7cyt6ZIlh32EwteuexabycNlE1puVLkICxQC/aN43Qme7cZg
-----END RSA PRIVATE KEY-----
```

Note, that *ec2-user* user has sudo rights on each server.

As mentioned above, application code was stored in the Git bare repository on the ***git-01*** server. Unfortunately we were not able to restore this server or the repository. We managed, however, to find a working copy of the code (see this file, netent-slot.tar.gz). All the history, however, is gone and some build scripts are missing as well. In the README.md file there are some instructions left by developers.

Task 1: restore bare `slot.git` repository on the ***git-01*** server, so that developers could use it (clone/push/pull) as before.

Task 2: restore missing build scripts or create your own so that we can build distribution RPM (more information is in the README.md).

Before the incident, application was build and deployed using Jenkins pipeline on the ***jenkins-01*** server. It had two phases:

- ***build*** phase, to create distribution RPM package.
- ***deploy*** phase, to deploy created rpm to the ***static-01*** server.

Jenkins pipeline also used *jenkins_job* user (password: 12341234) to access `slot.git` repository on the ***git-01*** server via ssh. Private key for the *jenkins_job* user is :

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAqhYDY5NTj9P7gNmjW8FEgl6L6sMdqMpfuQGTXeGluItlRun/
UOP/2FFVmbVp6gP/tBZiTR3N99BBmL2L53h7s/7wCUCma2dcWeA62QjeQNkTcpms
nvJFnQqPZfHMQMdoE9isVpI0JQLiLBD3Si/Qz7aiAXDD9kq8xdHQq7umkC2a77fN
TZL/YDc1JKAwLioel4i1Bumoes9RDhh89ymcFg+7Hi8NeNRW4vKFRBzP54OywIBf
rrvnJPw41t42BLnmL2dactDq6grobfjIOMZheHY4RSCmg7DLdQVb/iS6SVVthrWV
+0A0YilEgWp4ivqp03jsEt202V3lAW/U5Jl1aQIDAQABAoIBAEWNmvUtWg6G8WJP
na336rwlTbJC9+3+VAajzk+l5s8Asfm91tqNYiJjoE2rEhTQbdCHFsvSVFfyoeCE
IqIVcwC4UBB4j5V8PxfByk7WgYyjOIC0txKIoE8H8nwN1SjoYnc7M75zKz+3XiYH
ChkcVJecZ8OPCZO+HFQI6PxRUFvEn8sHGVF6PIIlNS2ab68CxsBnEmwrjCJSqjqw
Ht355rVjdMmDNG5obtBeadBFMew01X1gWV+Cy5Yd1ayfj9yKhezt6u0N/cXHXJC5
8PgJ9N5TFQlKHzmY6Vy8wHEnqpjEwgkRmN0Nmpw53NrBfFqjnguTlaeGfK9HuqqQ
Bg+BvsECgYEA1tZYRYpDRTehRqCkD3Jxe91Sj1NeGjOPsx1PizrKYzyYTrYZpKV1
WZ12xOg3P1xSEK9crkS9/1ugdGkBiLoI61GbLgOv93r+WZPIKIsNHo9Lu/3hdVS/
0xsbpb7XYeKgDV1uWRhtH9iNDe55DeZ7KCkl+qGMx0zD5C3rG4KYEXMCgYEAyqyk
HLIo7RZhvqauqoA+nhyomCT3/WkJZKcE6uBbgl7Lie7O+MgXVfwmEOqAhOoxdssV
uHIg859kNqaDm3J7dr68NB+4xCvns8LlkK6JTOs6zonkLGjw8R0twDOb8Ynd7T/c
/5yccdOKPqFTUQDPqZAsarArIyO2y0/TH3tNNrMCgYEAn2u/ZRsMgDifvMFH/Q53
l+rMcP70PY01fTyq4i5mMO2uN/p5XOxLAI0QrV2EYcgoPaljpQ3tz6g2kSu6r7Md
G0IbXTHvSj5CsEYxqaRFv/+lu4LR4zdYgVcEVSIgwcph6okCcarmj1MqmMP1oJL/
oPWn8SwgQ2R7mL9aT7fBEHECgYEAoOXFs3JMw0uNdYlmCL2e+G7P+Vj/9xG3/0wI
v141x53yACKx9RcET6LJpYxFUz+I3UIEkFUqi73yFODEgle0ZGSvjMHSMQqZ+z+y
d9XX8lScbvNRTWBSR1hexcMy/eHxaTPWAqcWG9dQXMA+IISAV6C5YklRNahYqbdJ
aCi+97kCgYEAuzAZ2EGvOMZ0jGaplzdhzE1gfr3Bz7G5YW/g5dl+PbjdSwurNk2b
fVgH16c71e9WChGoCcn0fB1xogLipisltzROkTkSw/AiQytdr/VOJBg+43QbqBao
XuYaDkkjOTkWsM+nICBfF6HZnldvu/Spn2DjahSVEvJgvxYiX13KjM8=
-----END RSA PRIVATE KEY-----
```

The last successful deploy is still running on the ***static-01*** server, http://<static-01-ip>. Unfortunately Jenkins pipeline job as well as it's configuration is gone.

Task 3: restore Jenkins pipeline on the ***jenkins-01*** server so that:

1. Pipeline is triggered every time commit is pushed to `<git-01>:/srv/git/slot.git;`
2. Pipeline builds distribution RPM package;
3. Pipeline deploys new RPM package to the ***static-01*** server, so the new application version is available on http://<static-01-ip>**.**

Task 4: add functionality to the pipeline to ensure traceability and reproducibility of the created RPM (i.e. so that it is possible to trace installed on ***static-01*** RPM to the corresponding commit in Git repository).

Good luck!