

## **Git Commands with Technical Guftgu by Bhupinder Rajput**

### **Scenario :-**

**Launch one linux ec2 instance in Mumbai (ssh&http) region :-**

- 1) yum install git -y
- 2) which git
- 3) git --version
- 4) git config --global user.name "bhupinder"
- 5) git config --global user.email "[bhupinder@gmail.com](mailto:bhupinder@gmail.com)"
- 6) git config --list
- 7) mkdir mumbaigit
- 8) cd mumbaigit
- 9) git init
- 10) cat >mumbai1

dil e nadaan tujhe huaa kya hai

**ctrl+D**

- 11) git status
- 12) git add .
- 13) git status
- 14) git commit -m "My first commit from mumbai"
- 15) git status
- 16) git log
- 17) git show **3d94671af5e0b1e0676e9716e348fcf1c2b47582**
- 18) git remote add origin <https://github.com/akashbkochure/technicalguftgu.git>
- 19) git push -u origin master

Gitusername- \*\*\*\*\*

Gittokenpassword- \*\*\*\*\*

20) cat >mumbai1

aakhir is murj ki dava kya hai?

21) cat mumbai1

22) git status

23) git add .

24) git status

25) git commit -m "My second commit from mumbai"

26) git status

27) git log

28) git show **54e94320cc107d68f73b52cb91e5458cf0bee0e6**

29) git push origin master **or** git push -u origin master

Gitusername- \*\*\*\*\*

Gittokenpassword- \*\*\*\*\*

30) git status

### **\*Next Day**

1) git pull origin master

2) git log

3) git show ce742e93ef1166303f02125067cfdd3ec1d32ab0

4) cat .gitignore

\*.css

\*.java

5) git add .gitignore

6) git commit -m "ignore file format"

7) git status

8) touch file1.txt file2.java file3.txt file4.css file5.txt

9) git status

10) git add .

- 11) git status
- 12) git commit -m "ignore css and java files"
- 13) git status
- 14) git log
- 15) git show 3fa6e1e39583bfeb22241e6e9bc226403ae69bb3
- 16) touch Bhupinder.java
- 17) git status
- 18) touch Bhupinder.txt
- 19) git status
- 20) git log -1
- 21) git log -2
- 23) git log --oneline
- 24) git log --grep "ignore"
- 25) git log --grep "commit"
- 26) git push -u origin master**
- Gitusername- \*\*\*\*\***
- Gittokenpassword- \*\*\*\*\***
- 27) git status

### **Launch other linux ec2 instance in Singapur (ssh&http) region :-**

- 1) yum install git -y
- 2) which git
- 3) git --version
- 4) git config --global user.name "akash"
- 5) git config --global user.email "[akash@gmail.com](mailto:akash@gmail.com)"
- 6) git config --list

- 7) mkdir singaporegit
- 8) cd singaporegit
- 9) git init
- 10) ls -la
- 11) **git remote add origin** <https://github.com/akashbkochure/technicalguftgu.git>
- 12) git log
- 13) git status
- 14) git pull origin master
- 15) git show 54e94320cc107d68f73b52cb91e5458cf0bee0e6
- 16) cat >>mumbai1

hume unse hai wafaa ki umeed

jinhe maalum hi nahi wafaa kya hai

**ctrl+D**

- 17) cat mumbai1
- 18) git status
- 19) git add .
- 20) git status
- 21) git commit -m "my first commit from singapore"
- 22) git status
- 23) git log
- 24) git show ce742e93ef1166303f02125067cfdd3ec1d32ab0

**25) git push -u origin master**

**Gitusername- \*\*\*\*\***

**Gittokenpassword- \*\*\*\*\***

- 26) git status

**\*\*\* Next Day :-**

- 1) git pull origin master

- 2) git log
- 3) git branch
- 4) git branch branch1
- 5) git branch
- 6) ls
- 7) git checkout branch1
- 8) git branch
- 9) ls
- 10) git log --oneline
- 11) cat >branch1file

Hello to All, From my side (Akash Kochure),  
“Happy Christmas ! from to all by my side.”

- 12) git status
  - 13) git add .
  - 14) git status
  - 15) git commit -m "branch1 first commit from singapore"
  - 16) git status
  - 17) git log
  - 18) git checkout master
  - 19) git branch
  - 20) ls
  - 21) git checkout branch1
  - 22) git branch
  - 23) ls
  - 24) cat >branch1secondfile
- Hiiiiiiiiii
- 25) ls

- 26) git branch
- 27) git checkout master
- 28) git branch
- 29) ls
- 30) git checkout branch1
- 31) git branch
- 32) ls
- 33) git add .
- 34) git commit -m "second commit branch1 from singapore"
- 35) git log --oneline
- 36) git checkout master
- 37) ls
- 38) git log --oneline
- 39) git merge branch1
- 40) ls
- 41) git log --oneline
- 42) git status
- 43) git push -u origin master**
- Gitusername- \*\*\*\*\***
- Gittokenpassword- \*\*\*\*\***
- 44) git status
- 45) git checkout master **or** git branch (master)
- 46) ls
- 47) cat >rajputfile
- Hello Rajput
- 48) git add .
- 49) git commit -m "first commit before conflict"

50) git checkout branch1

51) git branch (branch1)

52) cat >Rajput

Hieee Rajput

53) git add .

54) git commit -m "commit from branch1"

55) git checkout master

56) git branch

57) git merge branch1

**It show conflict**

58) vi rajputfile

Change something in that file & save.

59) git status

60) git add .

61) git commit -m "resolve conflict"

62) git log --oneline

63) git status

64) touch demofile

65) git add .

66) git commit -m "demofile"

67) vi demofile

First code

**Save**

68) git stash

69) cat demofile

70) git stash list

71) vi demofile

Second code for client

### **Save**

72) cat demofile

73) git stash

74) git stash list

75) cat demofile

78) git stash apply stash@(1)

79) cat demofile

80) git add .

81) git commit -m "first code done"

82) git stash apply stash@(0)

### **Conflict**

83) vi demofile

### **Delete first code**

#### **save**

84) git add .

85) git commit -m "second code done"

87) git status

88) git log --oneline

89) git stash list

90) git stash clear

91) git stash list

92) **git reset :-** it is a powerful command that is used to undo local changes to the state of git repo.

93) git branch

94) cat >testfile

Hello Bhupinder

### **Save**



95) git add .

96) git status

97) git reset .

98) git status

99) git add .

100) git status

101) git reset --hard

102) git status

103) **git revert :-** It helps you undo an existing command.

104) cat >revertfile

Paneer code

105) git add .

106) git commit -m "this is first paneer code commit"

107) cat >>revertfile

Wrong Mushroom code

108) git add .

109) git commit -m "this is wrong mushroom commit"

110) git status

111) git log

112) cat revertfile

Paneer code

Mushroom code

113) git revert (mushroom log id)

**Change file.....please ignore previous commit.....save**

114) ls

115) cat revertfile

116) git log --oneline

117) touch filex filey filez

118) ls

119) git status

120) git clean -n

**Show you to confirmation about remove or delete file**

121) git clean -f

**Removing**

122) ls

123) git log --oneline

124) git tag -a important -m “ this is very imp commit” **Commit-Id**

125) git show important

126) git tag -d important

127) git tag

128) git log --oneline

