By default shell script can be commented out prefixing # character, for example:

```
# my comment goes here
```

Bash Shell Script Put Multiple Line Comment Syntax

For multiline comment use the following syntax:

```
#!/usr/bin/env bash
# my comment  1
# my comment  2
# my comment  N
```

However, you can use HERE DOCUMENT feature as follows:

```
#!/bin/bash
echo "Say Something"
<<COMMENT1
    your comment 1
    comment 2
    blah
COMMENT1
echo "Do something else"
```

This type of redirection tells the shell to read input from the current source (HERE) until a line containg only word

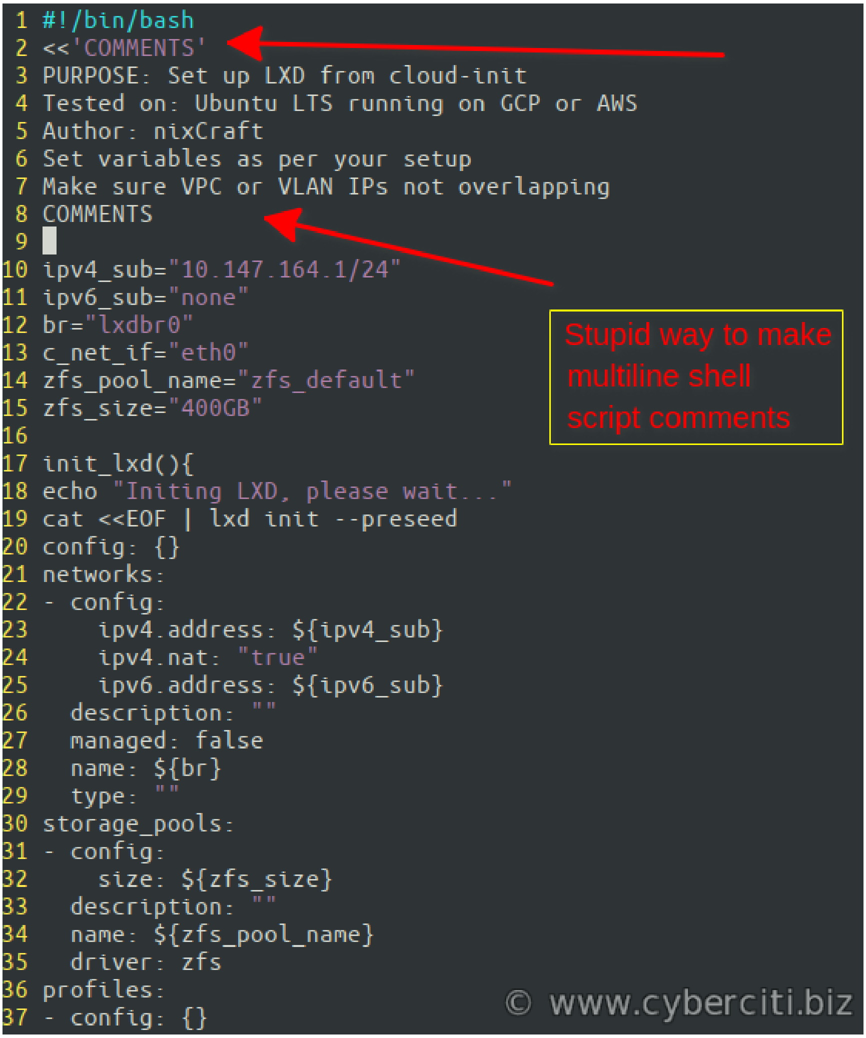
(HERE) is seen. HERE word is not subjected to variable name, parameter expansion, arithmetic expansion, pathname expansion, or command substitution. All of the lines read up to that point are then used as the standard input for a command. Files are processed in this manner are commonly called here documents. If you do not want variable name, parameter expansion, arithmetic expansion, pathname expansion, or command substitution quote HERE (COMMENTS) in a single quote:

```
<<'COMMENTS'
text1
text2
testN
$varName
COMMENTS
```

Bash comment block example

It is important that you put EOF in a single quote ('**EOF**') to avoid command execution and phrasing. Here is block comments in a shell script example:

```
#!/usr/bin/env bash
echo "*** Before comment block ***"
: <<'EOF'
CODE block starts
CODE block ends here
EOF
echo "*** After comments block ***"
```



```
#!/bin/bash
<<'COMMENTS'
PURPOSE: Set up LXD from cloud-init
Tested on: Ubuntu LTS running on GCP or AWS
Author: nixCraft
Set variables as per your setup
Make sure VPC or VLAN IPs not overlapping
COMMENTS

ipv4_sub="10.147.164.1/24"
ipv6_sub="none"
br="lxdbr0"
c_net_if="eth0"
zfs_pool_name="zfs_default"
zfs_size="400GB"

init_lxd(){
echo "Initing LXD, please wait..."
cat <<EOF | lxd init --preseed
config: {}
networks:
- config:
    ipv4.address: ${ipv4_sub}
    ipv4.nat: "true"
    ipv6.address: ${ipv6_sub}
  description: ""
  managed: false
  name: ${br}
  type: ""
storage_pools:
- config:
    size: ${zfs_size}
  description: ""
  name: ${zfs_pool_name}
  driver: zfs
profiles:
- config: {}
```

Writing multi-line comments in Bash scripts

Multiline shell script comments

Another option as pointed out by Ikram in the comments section below:

```
#!/bin/bash
foo=bar
: '
This is a test comment
Author foo bar
Released under GNU
'

echo "Init..."
# rest of script
```

Please note that the `:` is shell builtin command. From the bash(1) man page:

> : [arguments] No effect; the command does nothing beyond expanding arguments
> and performing any specified redirections. A zero exit code is
> returned.

So the syntax is:

```
: '
 your comments here
'
```

Null command used in your shell script to put multiple comments. The '`:`' has no effect. In other words, the command does nothing and it always exit with succeed code.