Generate two SSH Keys:
Mission-Systems (GitLab - tariq.abuhashim@missionsystems.com.au):
`ssh-keygen -t ed25519 -C "you@company.com" -f ~/.ssh/id_ed25519_gitlab`
Personal (GitHub - t.abuhashim@gmail.com):
`ssh-keygen -t ed25519 -C "you@personal.com" -f ~/.ssh/id_ed25519_github`
You will have:

- `~/.ssh/id_ed25519_gitlab` and `id_ed25519_gitlab.pub`
- `~/.ssh/id_ed25519_github` and `id_ed25519_github.pub`

Add keys to SSH agents:
Start the agent (if not running):
`eval "$(ssh-agent -s)"`
Then add the keys:
`ssh-add ~/.ssh/id_ed25519_gitlab`
`ssh-add ~/.ssh/id_ed25519_github`

Add keys to SSH agents:
Edit SSH config file:
`nano ~/.ssh/config`
Add the following content:

```
# GitHub (personal)
Host github.com
  HostName github.com
  User git
  IdentityFile ~/.ssh/id_ed25519_github
  IdentitiesOnly yes

# GitLab (work)
Host gitlab.com
  HostName gitlab.com
  User git
  IdentityFile ~/.ssh/id_ed25519_gitlab
  IdentitiesOnly yes
```

Add the public keys to the right Git Accounts:
GitHub: Paste the contents of `id_ed25519_github.pub` into a new GitHub SSH key window
`cat ~/.ssh/id_ed25519_github.pub`
Gitlab: Paste the contents of `id_ed25519_gitlab.pub` into a new GitLab SSH key window
`cat ~/.ssh/id_ed25519_gitlab.pub`

Test the Keys:

```
ssh -T git@github.com
# should say: "Hi username! You've successfully authenticated..."
ssh -T git@gitlab.com
# should say: "Welcome to GitLab, username!"
```

Tell Repo Locally who you are:

```
cd repo_dir
git config user.email "t.abuhashim@gmail.com"
git config user.name "Tariq-Abuhashim"
```

Use correct Git Remotes: personal (t.abuhashim), mission-systems (t.abuhashim)
Make sure the Git remotes use the SSH format, like:

```
git@github.com:t.abuhashim/repo_name.git
git@gitlab.com:t.abuhashim/repo_name.git
```

Check and set remotes

```
git remote set-url origin git@github.com:Tariq-Abuhashim/repo.git
git remote set-url origin git@gitlab.com:missionsystems/repo.git
```

Verify remote:

```
git remote -v
```