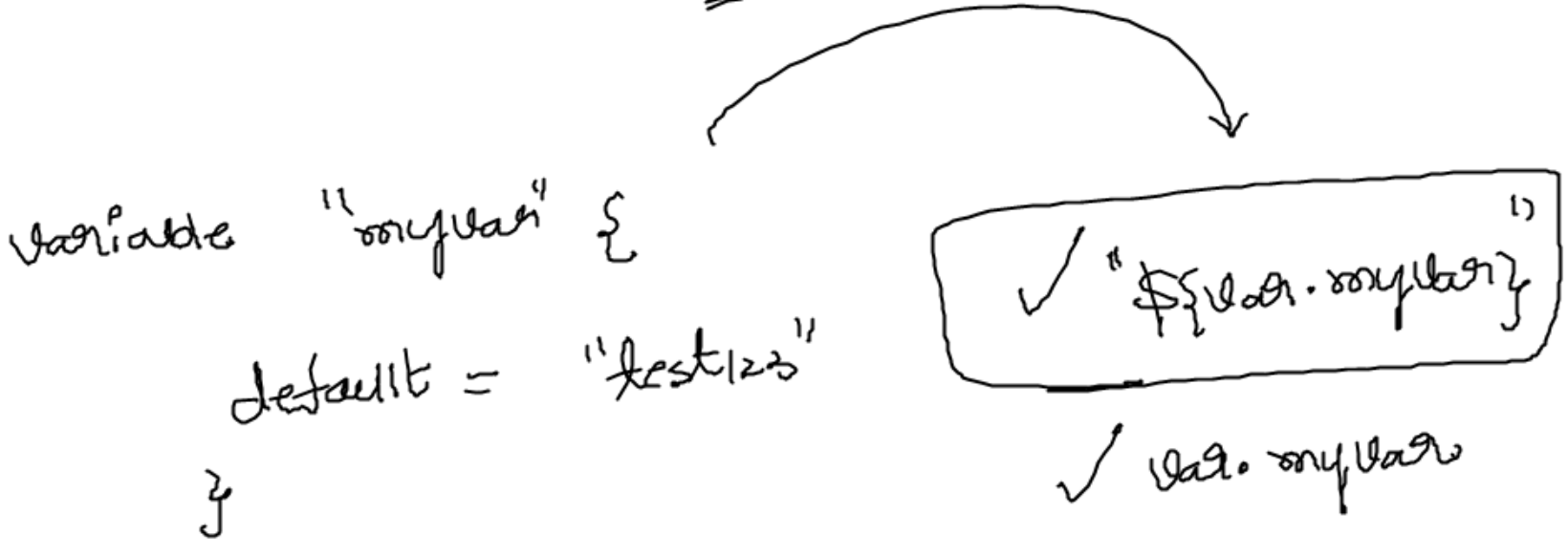
Terraform

modules

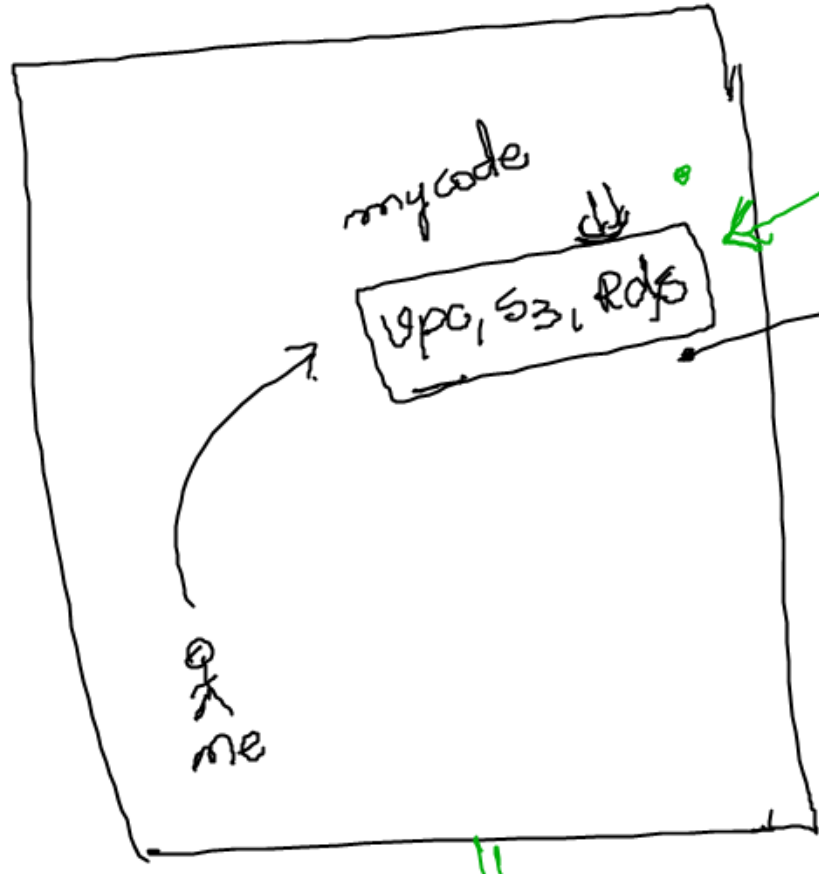
0.11

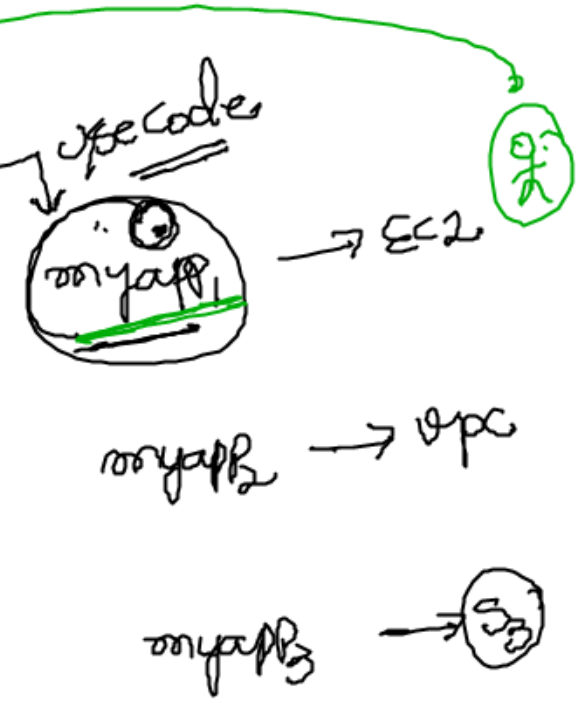
0.15

download

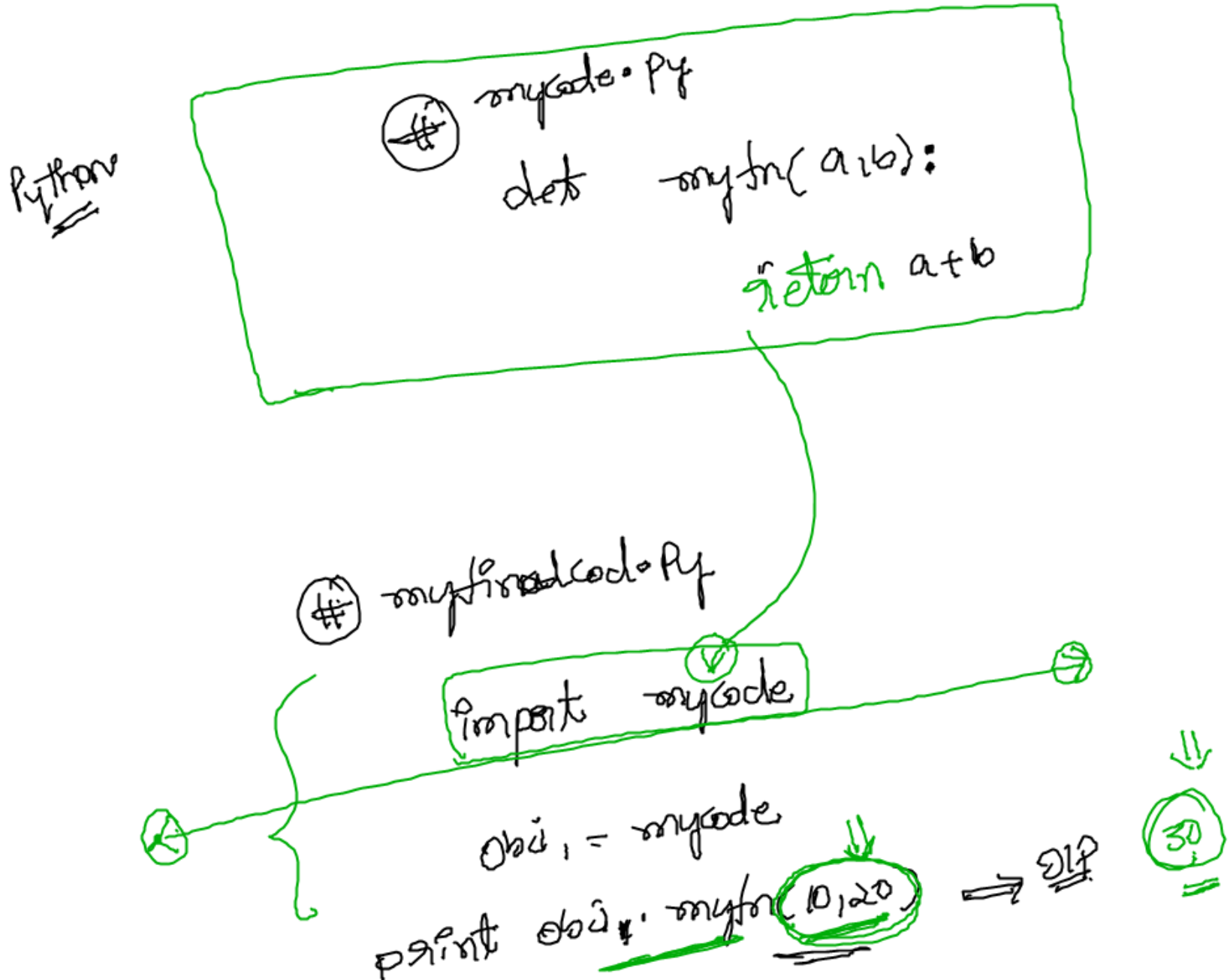


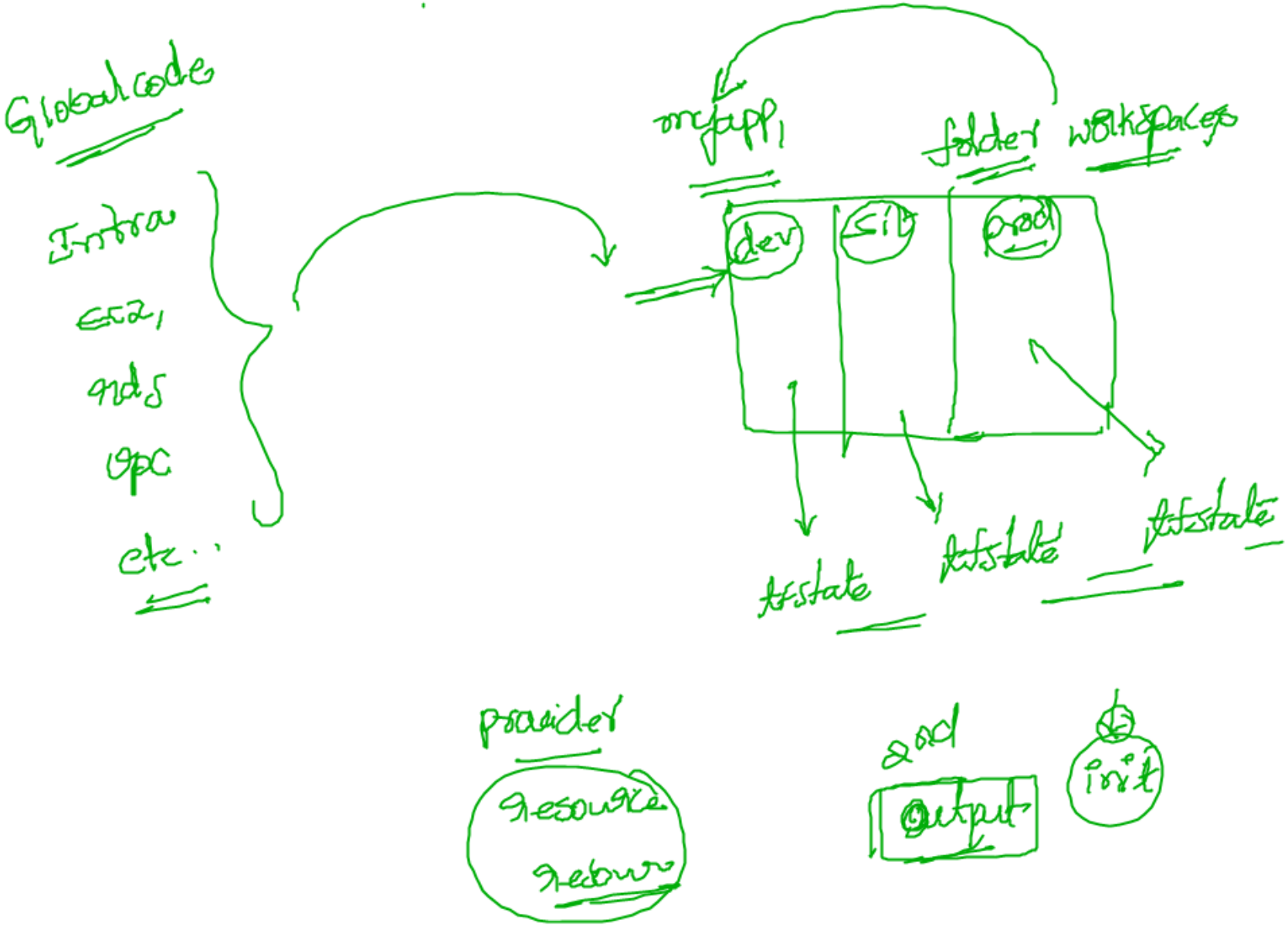
Reusability




```
module "mymodulename" {
  source = "pathtomodules"
  arg1   = "value1"
  arg2   = "value2"
}
```


Python
mycode.py
def myfn(a,b):
return a+b
myfinalcode.py
import mycode
obj1 = mycode
print obj1.myfn(10,20)
30


Global code
Intra
ec2
rds
vpc
etc..
my app
folder workspaces
dev
sit
prod
tfstate
tfstate
tfstate
provider
resource
resource
data
output
init

Managing infra

provisioning

Managing

init ⟹ download and load resources binaries

apply ⟹ create infra

plan ⟹ it will be a quick check (dry run)

destroy ⟹ destroy infra

```
def mydata(a=20, b=30):
    print a+b
```

| | | O/P |
|---|---|---|
| mydata() | → | 50 |
| mydata(50) | → | 80 |

-var

variables "a" {

⇒ ~~default~~ ⇒ 20

}

| | | O/P |
|---|---|---|
| "${var.a}" | ⇒ | 20 |
| "${var.a}" | ⇒ | 40 |

⊙ myvars.tfvars

a=40

overwrite variables values

$\Downarrow$

$\boxed{-var}$ varname = $val_1$ ✓

30

-var " = $val_2$

-var " = $val_3$

$\Downarrow$

• Avars

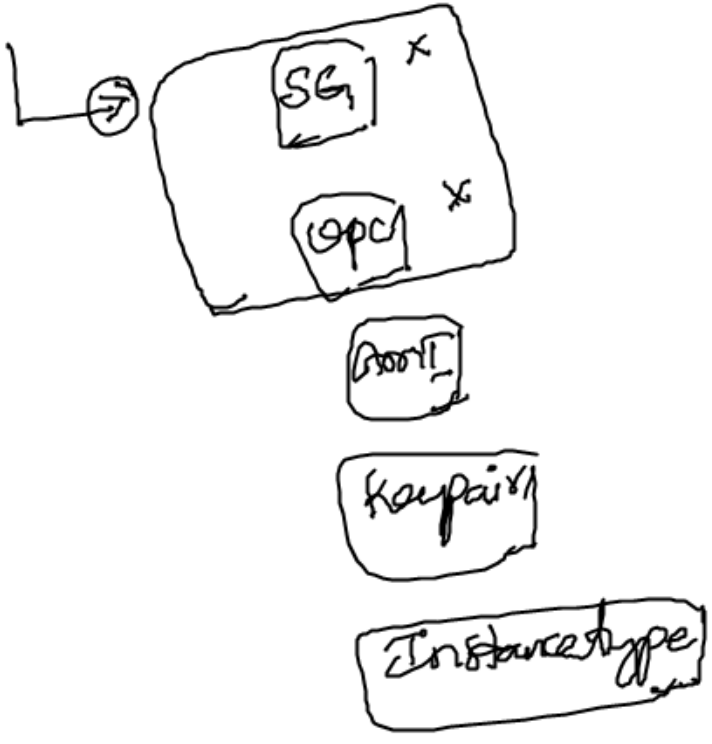
$var_1 = 10$

$var_2 = 20$

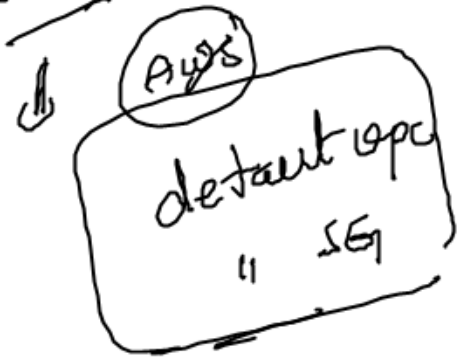
$3 = 30$

$30 = 300$

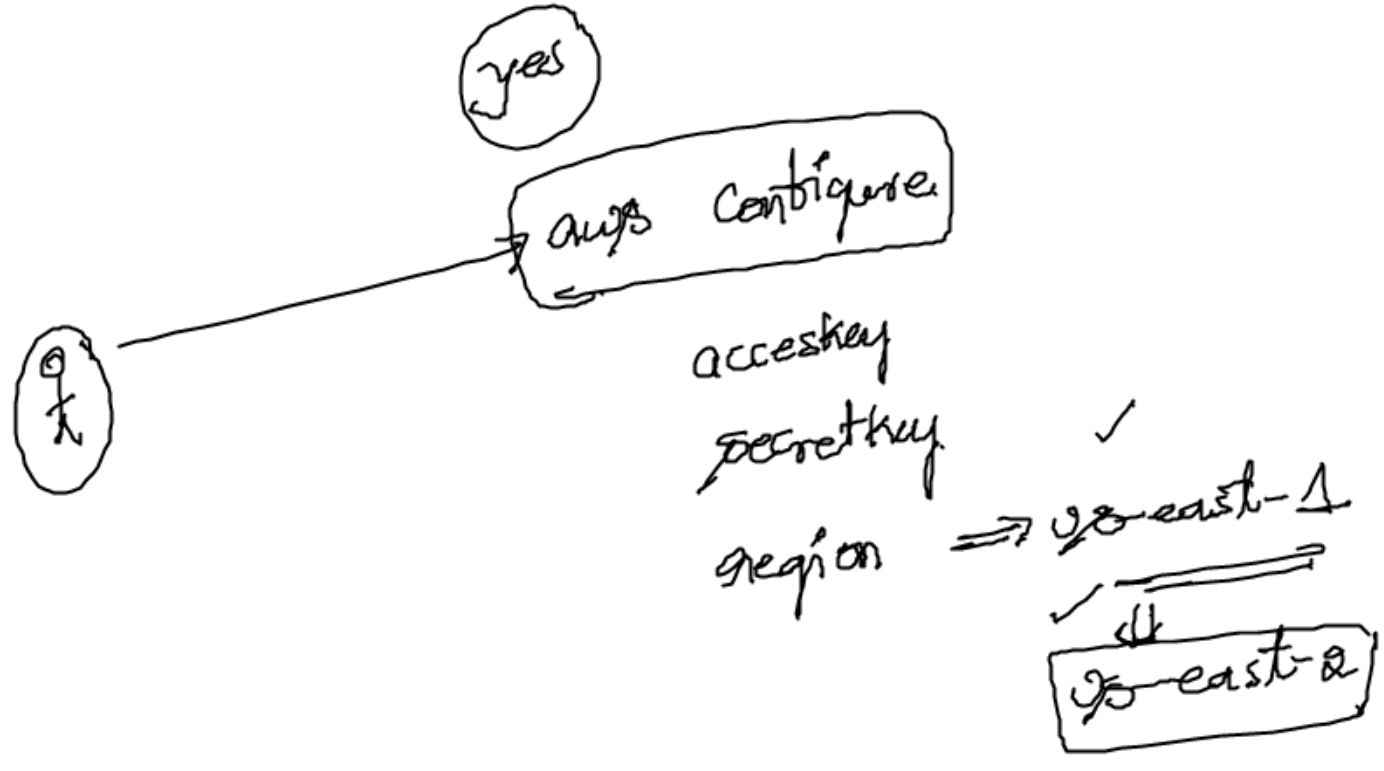
EC2 creation

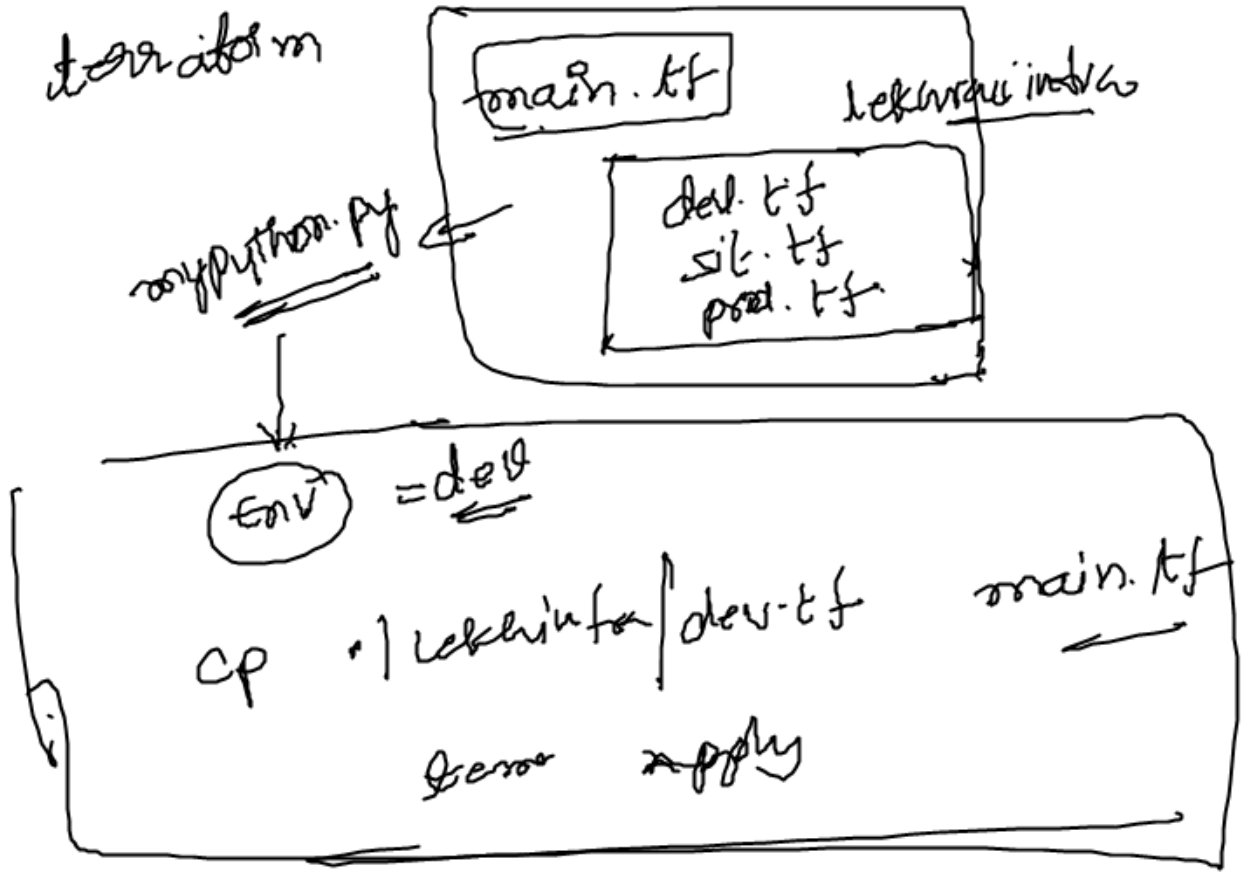



default's



profiles


yes
aws configure
accesskey
secretkey
region
us-east-1
us-east-2


terraform
main.tf
lekhrani infra
dev.tf
sit.tf
prod.tf
mypython.py
Env
=dev
cp
lekhinfra
dev.tf
main.tf
terra
apply