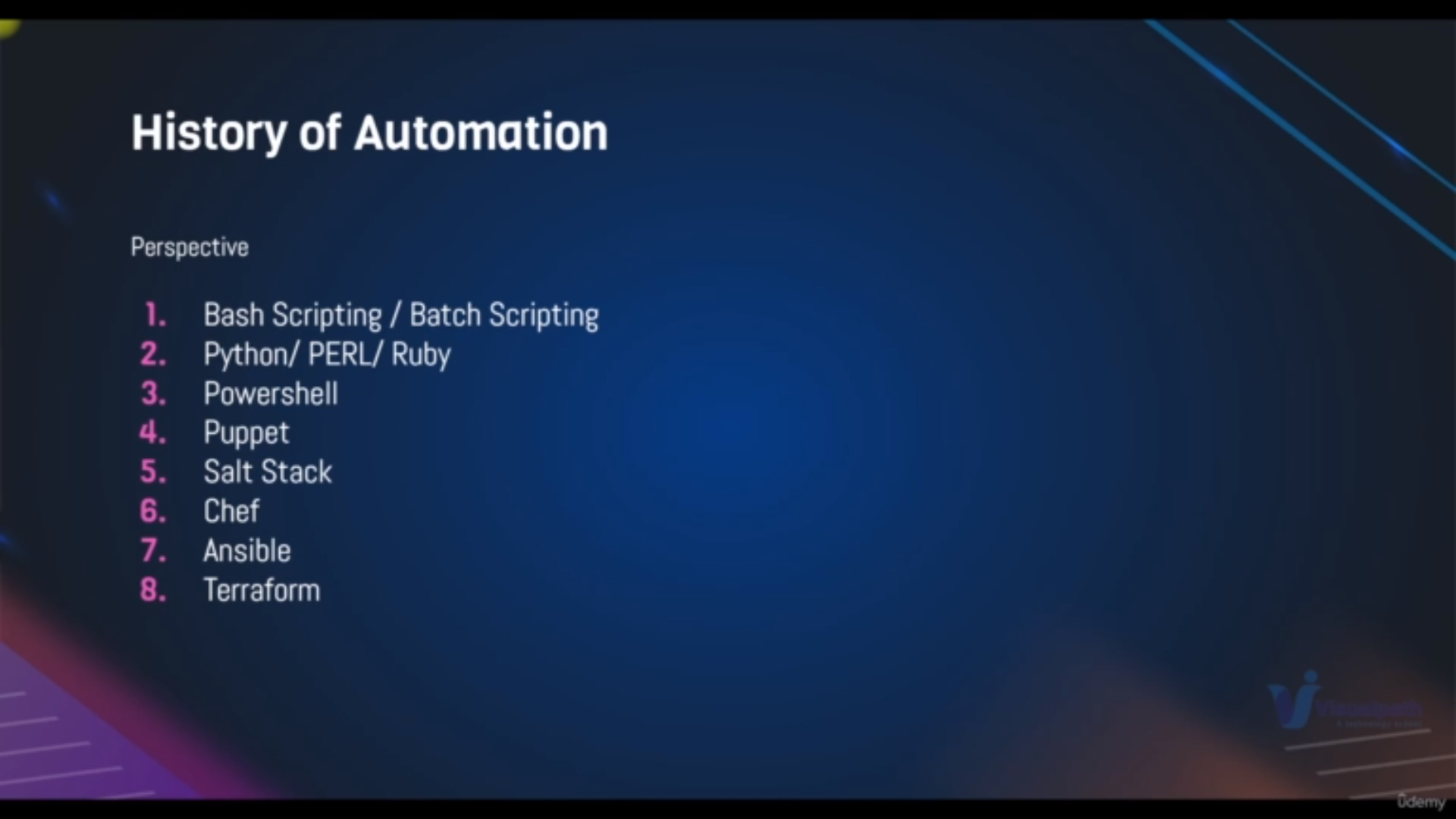
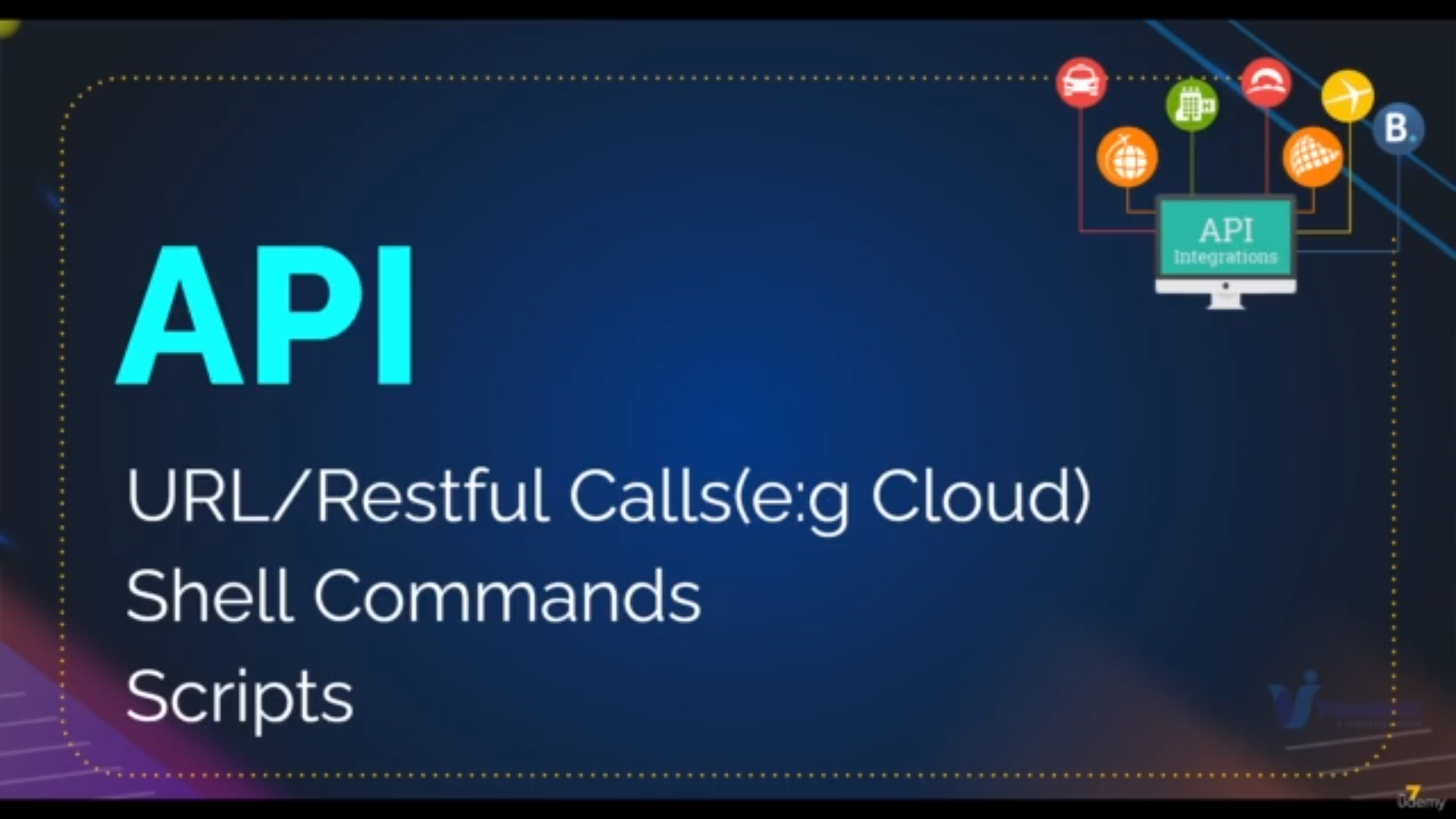
**Ansible:** to manage the code infracture and change the configuration

scripting→ configurations

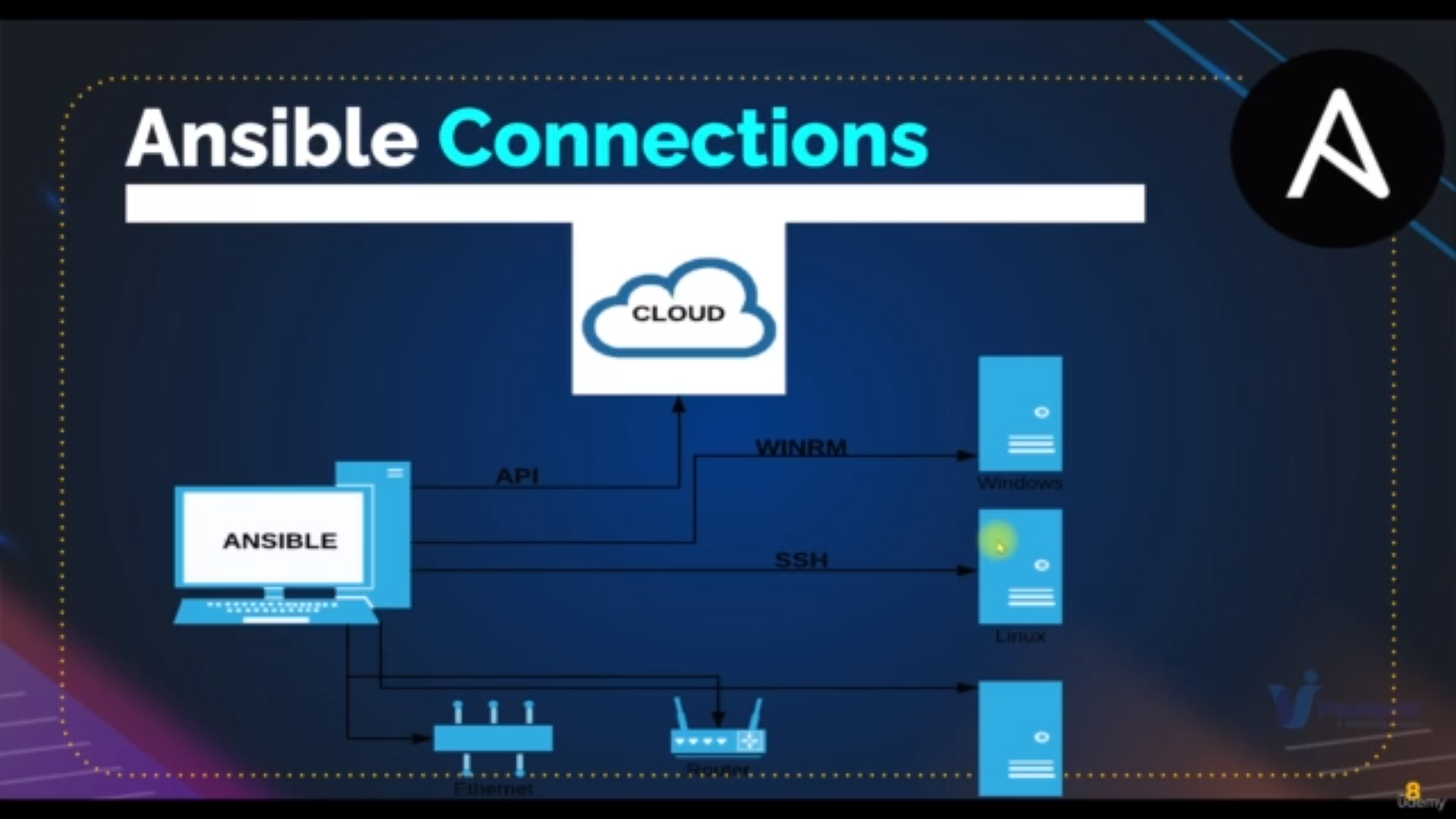
uses of ansible

ansible connect through ssh connection in linux

ansible needs yaml not programming language

ansible uses lot of api like

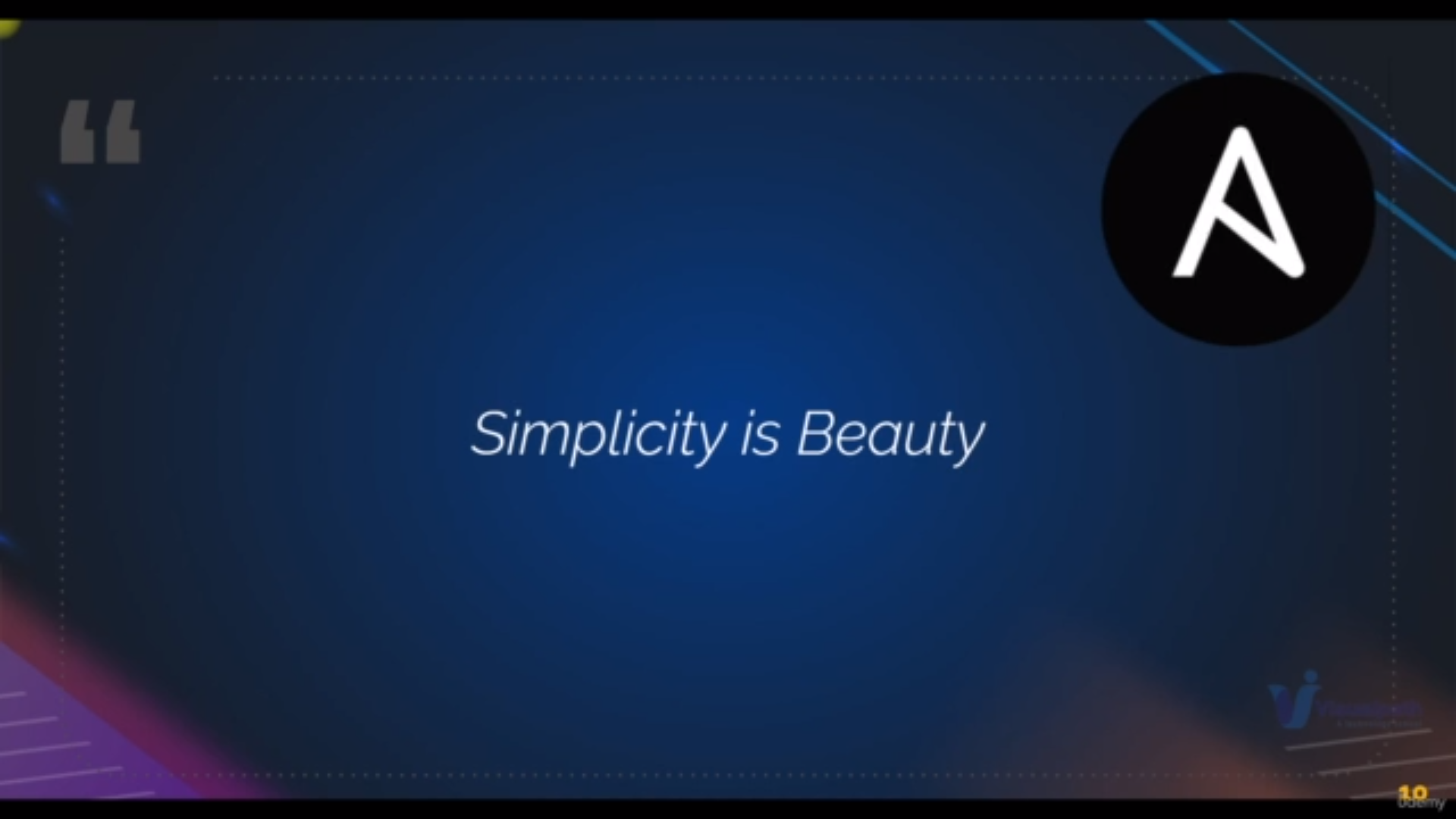
bottom api is used for aws

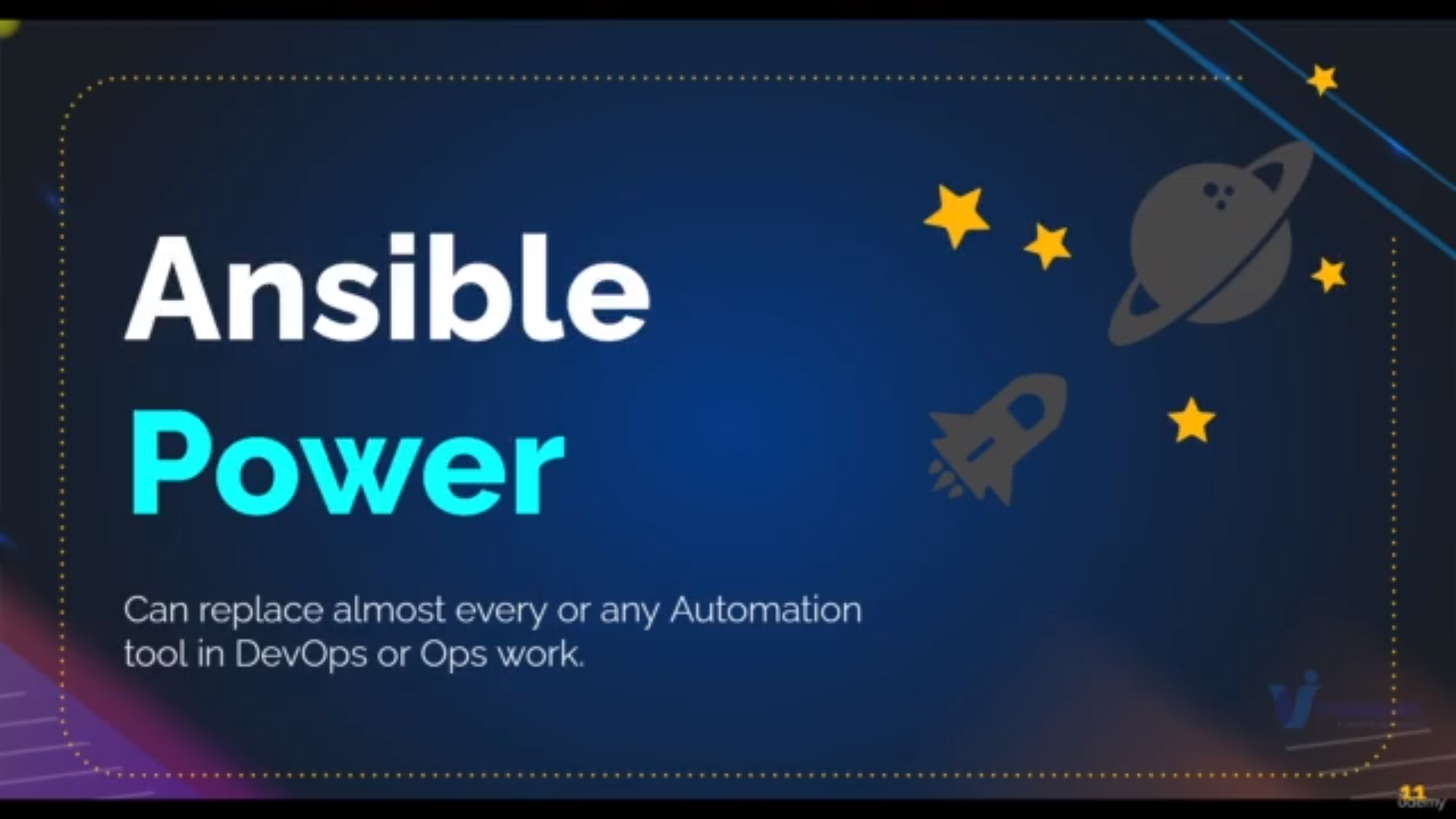
ansible also used to manage network device like switches and routers

we have ipaddress in inventory file of ec2 machines and modules

we’ll write playbooks using yaml where we use modules and excute on ec2machines which are there in inventory file

when we are excueting playbooks these are converted into python scripts and execute

ansible is simple to use

ansible has lots of features

**Setup for Ansible:**

**1) launching instances**

a) launch a ubuntu instance with General configuration as **control machine** and

add the user data as

---------------------------------------------------------------------------------------------

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 93C4A3FD7BB9C367

sudo apt update

sudo apt install ansible -y

----------------------------------------------------------------------------------------------

b) launch two instances as **web01,web02** with General configuration

c) launch one instances as **db01** with General configuration

2) **Making ssh connection**

After installing ansible through above user data

we have to connect all our instances to contorl-machine through ssh-connection

for this we have to in invetory file [by default we /etc/ansible/host is an inventory file] we are overirding it

**inventory file** is the file where we will give all information of instances or hosts

so that we can ssh connection

it look like

-------------------------------------------------------------------------------------------------------------

web01 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

web02 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

db01 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

-----------------------------------------------------------------------------------------------------------

we can also groups

-------------------------------------------------------------------------------------------------------------

web01 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

web02 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

db01 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

[websrv]

web01

web02

[dbsrv]

db01

-----------------------------------------------------------------------------------------------------------

we can also make group of groups

-------------------------------------------------------------------------------------------------------------

web01 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

web02 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

db01 ansible\_host=private\_ip ansible\_user=ec2-user ansible\_ssh\_private\_key\_file=file.pem

[websrv]

web01

web02

[dbsrv]

db01

[servers:children]

websrv

dbsrv

-----------------------------------------------------------------------------------------------------------

so like we can make groups in inventory file

Note:**file.pem**  should there in directory where the invetory file located

then execute the command :

ansible -i inventory -m ping all

it will show the connections are established or not