Ansible Control Node requirements:

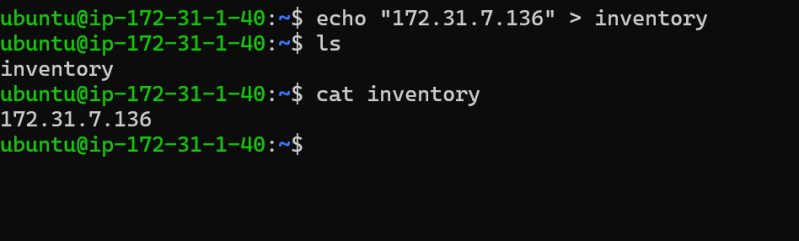
* + OS:
    - Linux flavours => RedHat, Ubuntu, CentOS, any of BSD’s
    - Mac
  + Software:
    - Python 2
    - Python 3
* Let’s create a ubuntu 18 VM as node
* Ansible Node requirements

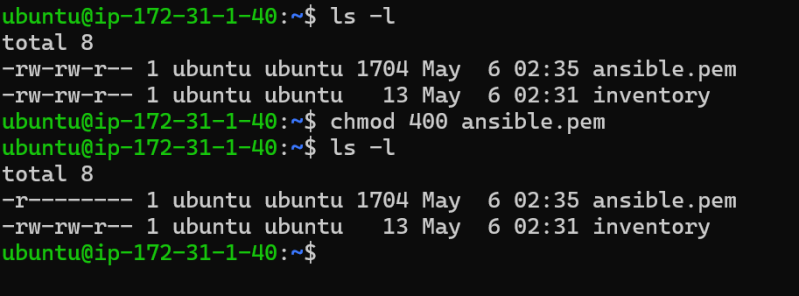
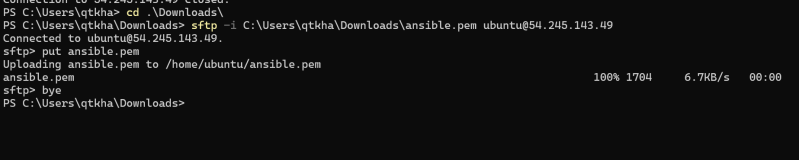
SSH should be running 

* + Python should be installed
  + SFTP/SCP should be configured
* Now let’s try to verify the connectivity using ansible b/w control node and node
* First let’s create a simple inventory

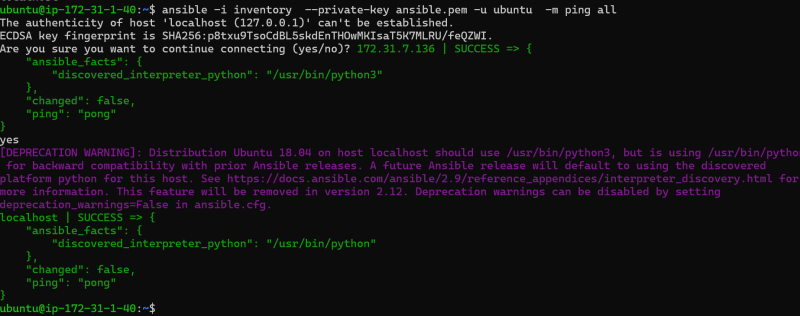
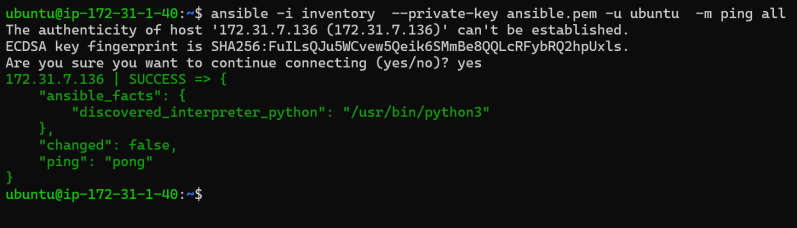
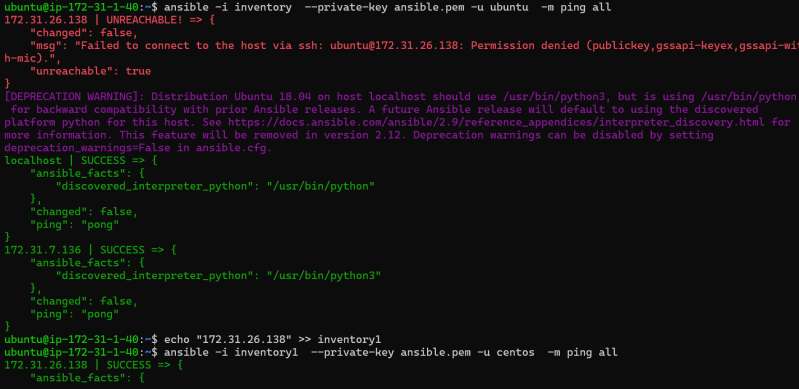
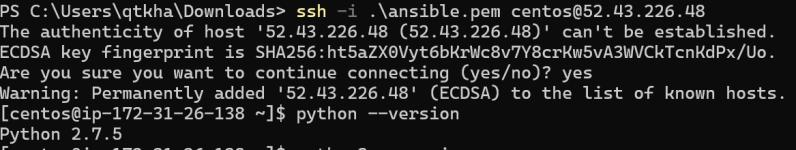
172.31.7.136

# echo "172.31.7.136" > inventory

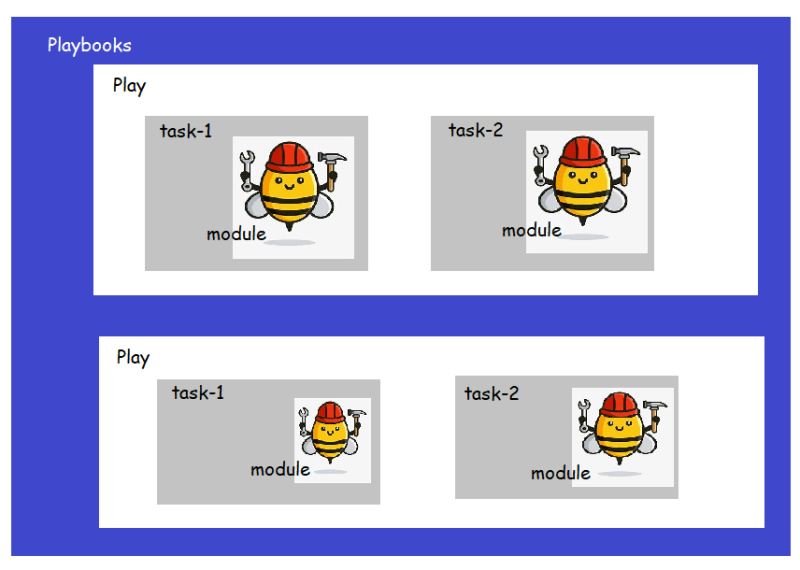


* Now ensure you have private key (pem) file in the Ansible control node and permissions are set 
* Let’s look at a simple ansible ping command which verifies if your control node can access nodes or not

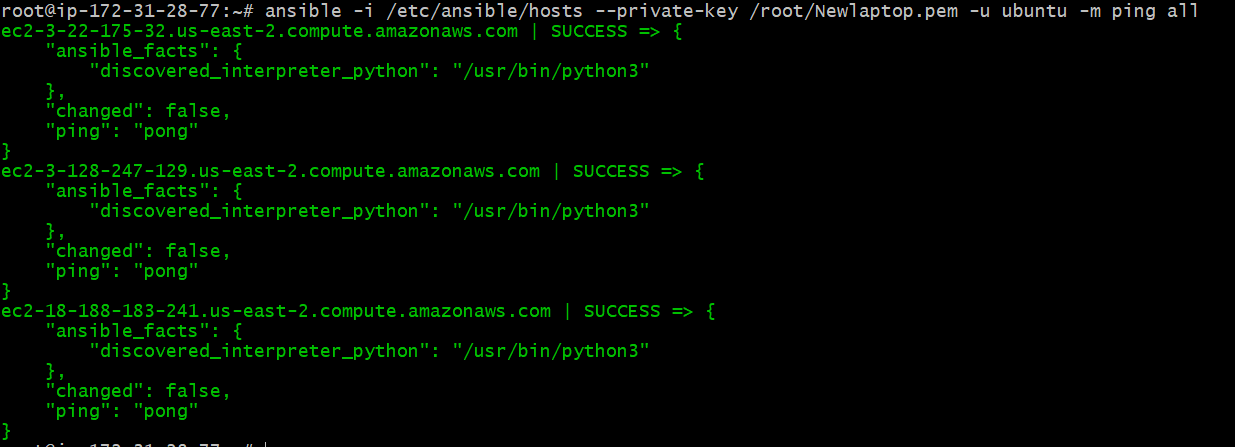
ansible -i <inventory> -m ping all

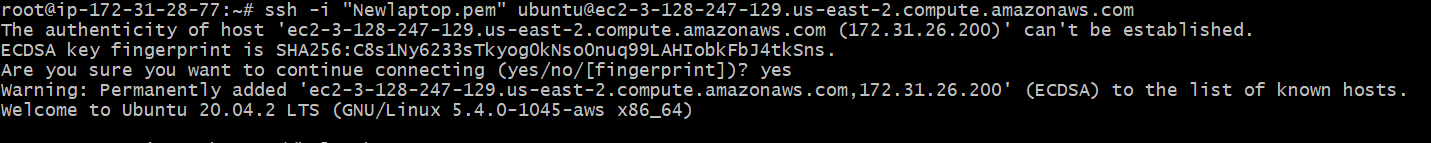
* We also need to specify the username and pem file (private key file) 
* So now lets add a new centos 7 VM 

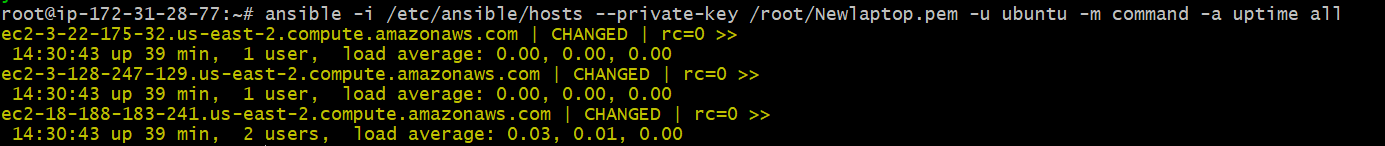
**How to make ansible work towards our deployments**

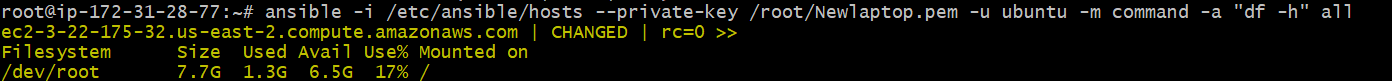
* Pictorial Representation 
* In Ansible the atomic unit of work is module. Ansible has lot of inbuilt modules to help us automate our infra needs
* Our approach to use Ansible
  + Make a list of all the steps needed to deploy your application
  + for every step try to find the ansible module which can help in expressing a desired state.

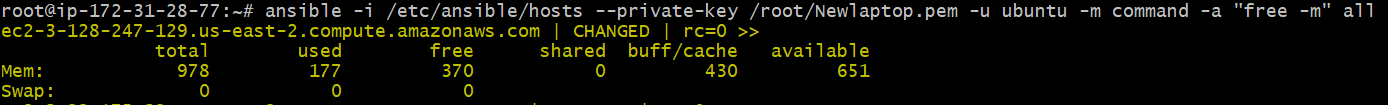
My practice output: Before running ansible first do ssh connectivity test.





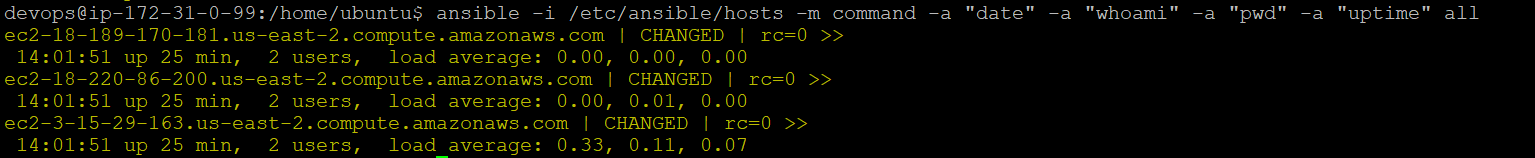








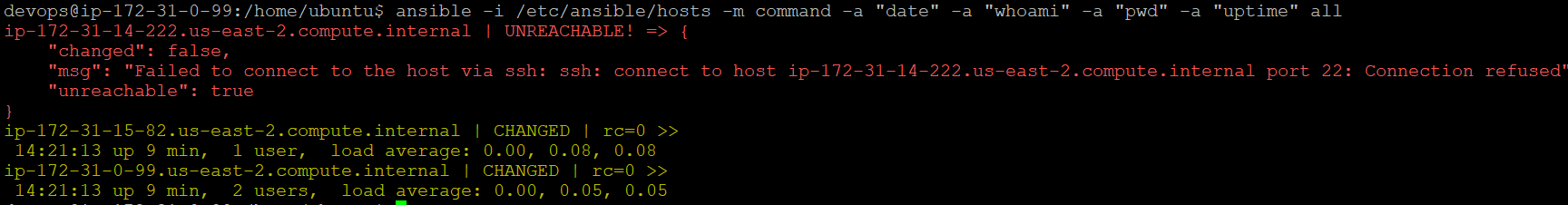






Stop sshd into slave node.

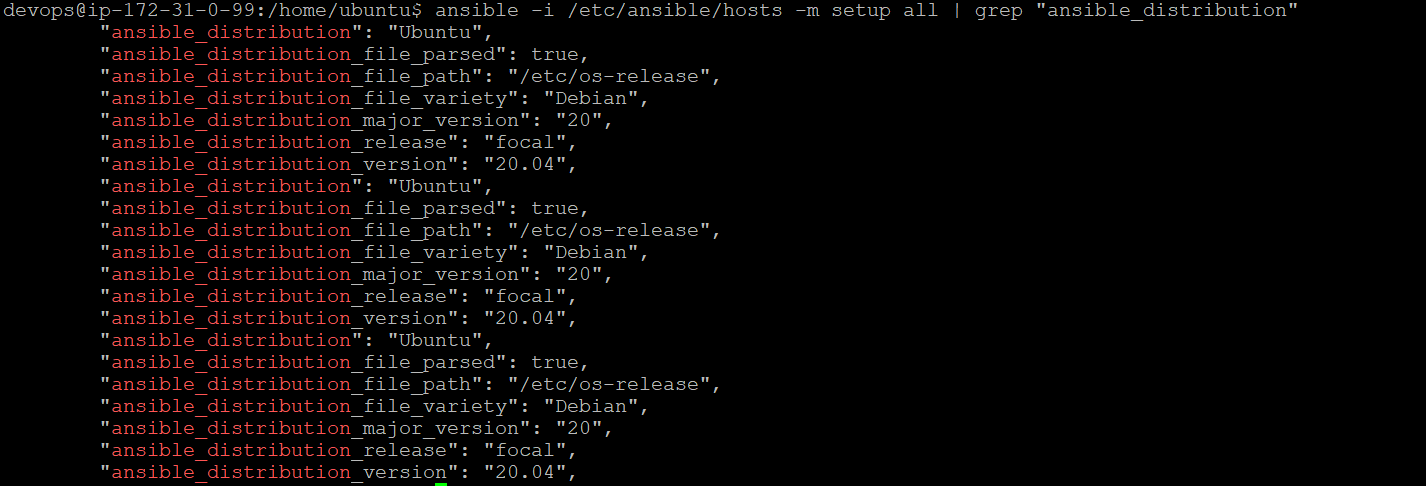




**Modules in Ansible**

Ansible setup module: This module will have all OS related information such as distribution, IP and host details.

ansible -i /etc/ansible/hosts -m setup all | grep "ansible\_distribution"



Link: <https://docs.ansible.com/ansible/latest/collections/ansible/builtin/setup_module.html>

Shell module: This module used to execute the shell commands.

Link: <https://docs.ansible.com/ansible/latest/collections/ansible/builtin/shell_module.html>

APT/YUM/Package module: This are for installation purpose.

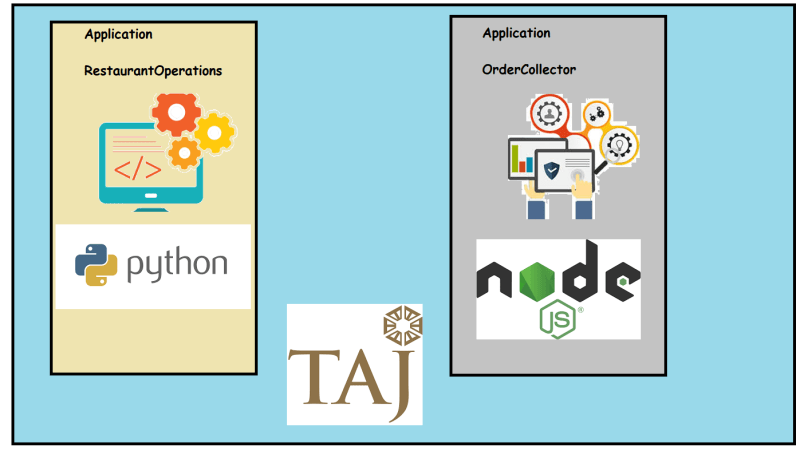
Debug module: for printing messages.

Copy module: Source, destination, permission, mod.

File module: Creating file and working on it.

**YAML (Yet Another markup language)**

**Story of Taj Group (Fictional)**

* Taj group has two applications from different vendors 
* How can two applications exchange data?
  + We might need a data exchange format which
    - is easier to understand
    - No particular programming languages assumptions
    - Easy to read (by both humans & programs)
* Lets Start with XML: Two applications should agree to a structure. This can be can be defined by xsd

<Order>

<from>Swiggy</from>

<placedon>01/10/2020 18:00:00</placedon>

<items>

<item>Biryani Family Pack</item>

<item>Coke </item>

</items>

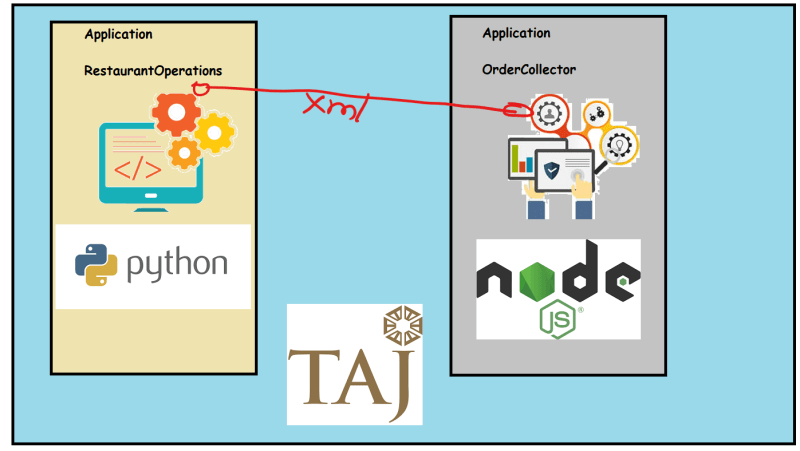
<delivery>

<placedon>01/10/2020 18:00:00</placedon>

<scheduledby>01/10/2020 18:15:00</scheduledby>

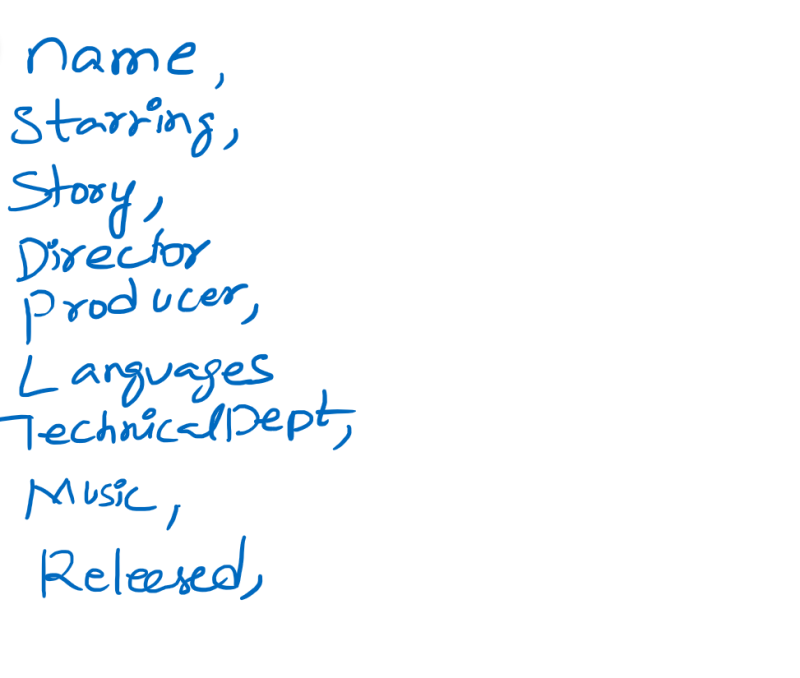
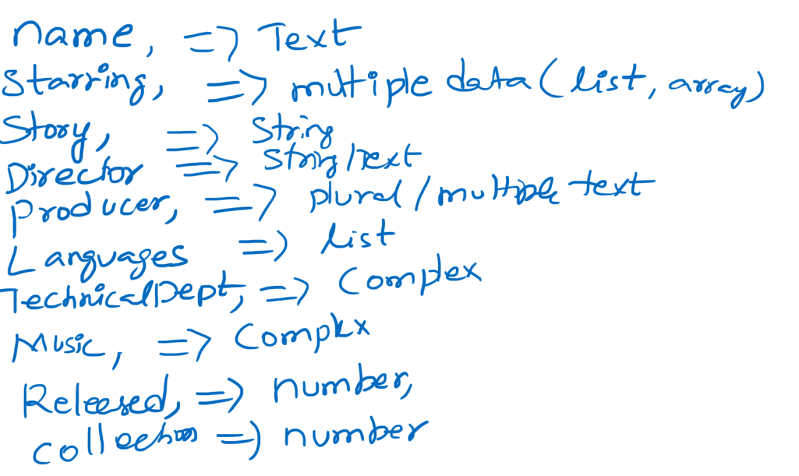
</delivery>

</Order>



* + XML as Data Exchange Format is very widely used, but parsing xml is difficult and sizes (payload) of the xml are more in size

**Json (JavaScript Object Notation)**

* Data is represented as name value pairs
* Advantages:
  + It is light weight
  + Easy to read and write
* Data can be in the form of
  + Text
  + Numbers
  + Boolean
* Data can be plural or complex
* Lets try to Think of information, in terms of name value pairs. Lets try to write the info about Bahubali
  + Different Name value pairs 
  + If we want to enter data in to the name value pairs the data types could be 
* Basic Json Reprsentation
  + In Json Data is represented as <name>: <value>
  + Eg:
  + "Hero": "Prabhas"
* Json Data Types: Value section might have different values. Value Types could be one of
  + String/Text: ”
  + Number
  + Boolean: true/false
  + list: represented in *[]*
  + object (Complex): Describing this type might require some more name value pairs. Represented in *{ }*
* Lets write a simple json to represent Bahubali

{

"title": "Bahubali The Begining",

"Director": "S.S. Rajamouli",

"Writer": "Vijayendra Prasad",

"BoxOfficeInCrores": 600,

"BudgetInCrores": 180,

"Released": "10/07/2015",

"Languages": [ "Telugu", "Tamil", "Malayalam", "Hindi" ],

"Distributors": {

"Telugu": "Arka Media Works",

"Tamil": [ "StudioGreen", "Sri Thenandal Films", "UV Productions"],

"Hindi": "Dharma Productions",

"Malayalam": "Global United Media"

}

}

**YAML (YAML Ain’t a Markup Language)**

* This also can be used for data exchanges and configurations
* Belives in indendation like python
* YAML is also collection of name value pairs
* In YAML Data is represented as <name>: <value>
* The Datatypes for Yaml are same as json.
  + Text is represented in quotes
  + List Items start with –
  + Object will start at the next indendation

Example of YAML file

---

name: 'Bahubali The Begining'

director: 'S.S. Rajamouli'

story: 'Vijayendra Prasad'

BoxOfficeInCrores: 600

BudgetInCrores: 180

Released: '10/07/2015'

Languages:

- Telugu

- Hindi

- Malayalam

- Tamil

Distributors:

Telugu: 'Arka Media Works'

Hindi: 'Dharma Productions'

Tamil:

- 'Studio Green'

- 'Sri Thenandal Films'

- 'UV Creations'

Malayalam: 'Global United Media'