**Ansible**

**Ansible (18 Note)**

**-------------------**

**shell disadvantages**

**1. shell script will not work in all linux distros.**

**2. scalability --> difficult to manage more no of servers**

**3. error handling and validations, need to manual**

**4. readability**

**5. shell will only work for linux, will not work with external systems**

**Configuration management**

**-------------------**

**configure your laptop**

**to make server ready for application deployment is called configuring server**

**deployment**

**------------**

**1. stop the server**

**2. remove old code**

**3. download new code**

**4. restart server**

**Idempotence -->**

**if you run a program multiple time, that can create same thing multiple times**

**even you run your program infinite times, it should not create any damage**

**providing same result irrespective of no.of executions is called idempotence**

**useradd ramesh**

**1. it will create multiple ramesh users --> bad**

**2. it will try to create user, but got error and program exit --> no**

**ansible --> if not exist it will create, if exist it will ignore --> yes**

**ansible-pull --> every 30min download the configuration from git directly and run it, anyhow we implemented idempotency so no probl**

**pull based configure**

**-------------------**

**chef server**

**we will configure agents to check every 1 hour**

**ansible-pull --> every 30min download the configuration from git directly and run it, anyhow we implemented idempotency so no probl**

**24\*7 = 168 time connect to server**

**bandwidth waste**

**resources consumption like CPU and memory**

**pull configuration have one advantage --> at the end of ansible**

**Linux --> everything is called as command**

**ansible --> module/collection**

**one by one module --> success/fail**

**keep all modules/commands in a file**

**keep some validations, conditions, loops, variables**

**now run the file**

**Playbook --> a file of modules/collections**

**YAML --> yet another markup language**

**DTO --> data transfer object**

**----------------------------**

**XML --> Extensive Markup Language**

**<User>**

**<Email>info@joindevops.com</Email>**

**<Password>admin123</Password>**

**</User>**

**JSON --> Java script object notation**

**{**

**"Emai": "info@joindevops.com",**

**"Password":"admin123"**

**}**

**YAML**

**----------------------------**

**email: "info@joindevops.com"**

**password: "admin123"**

**variables**

**--------------------**

**DRY --> don’t repeat yourself**

**variables we keep aside from the code, so that code will not be disturbed when we are doing changes.**

**variable holds a value, you can use it anywhere**

**data types**

**-------------------**

**list**

**map**

**boolean**

**key, value**

**conditions:-**

**id roboshop**

**if not exist we created, if exist we skipped**

**SOME\_VAR=$(command)**

**Diff bw shell & Command in ansible:-**

**command --> won't respect target machine shell variables and environment, it is running the command from outside**

**shell --> It is like you logged in inside target machine directly and running command**

**if I login to catalogue**

**-----------------------**

**dnf module disable nodejs -y ; dnf module enable nodejs:18 -y**

**"dnf module disable nodejs -y ; dnf module enable nodejs:18 -y"**

**Could not find or access '/tmp/catalogue.zip' on the Ansible Controller.\nIf you are using a module and expect the file to exist on the remote, see the remote\_src option**

**ansible controller/server --> catalogue**

**{**

**'changed': True,**

**'stdout': '11',**

**'stderr': '',**

**'rc': 0,**

**'cmd':**

**['mongo', '--host', 'mongodb.daws76s.online', '--quiet', '--eval', 'db = db.getSiblingDB(\"catalogue\"); db.products.count()'], 'start': '2023-12-22 02:39:19.967161', 'end': '2023-12-22 02:39:20.070547', 'delta': '0:00:00.103386', 'msg': '', 'stdout\_lines': ['11'], 'stderr\_lines': [], 'failed': False}**

**ansible.cfg --> ansible configuration file**

**1. ANSIBLE\_CONFIG (environment variable if set)**

**2. ansible.cfg (in the current directory)**

**3. ~/.ansible.cfg (in the home directory) --> it should be .**

**4. /etc/ansible/ansible.cfg**

**ansible vault:-**

**storage of secrets**

**asaiavaa --> devops**

**encoding --> a proper pattern to encode the text**

**aadaaeaavaaoaapaas**

**encryption with password --> AES256 (mathematic algorithm)**

**dfkgfagjg;gjgg --> devops**

**u4392854194586174598yhf --> devops**

**how to create ansible vault:-**

**ansible-vault create /path/some-name.yaml**

**dynamic inventory:-**

**we have servers, domains**

**we had on premise servers, apps are deployed manually**

**we used shell script to configure**

**2016 --> internet usage, digital revolution**

**2010/2012 --> EU, US**

**autoscaling of servers highly required**

**we are using ansible to configure**

**- if you have more servers dynamically**

**10 servers now --> because of traffic**

**I need to run ansilbe playbook against these servers**

**ansible --> AWS --> fetch ip address of the servers dynamically**

**fetch instances with name web in us-east-1**

**plug and play**

**if ansible have plugin to connect aws ec2, we can fetch ip addresses**

**boto and botocore --> aws python modules**

**ansible uses boto and botocore in background to connect aws**