

* What is Amazon CloudWatch?

Amazon CloudWatch is a monitoring service to monitor your AWS resources as well as the apps that you run on AWS i.e. it watches the servers & gives the reports.

Cloud Watch can monitor the:-

1. Compute services
 - a. EC2 instances
 - b. Auto scaling groups
 - c. Elastic Load balancers
 - d. Route53 Health checks
2. Storage & Content delivery
 - a. EBS volumes
 - b. Storage gateways
 - c. Cloudfront
3. Host level metrics consist of
 - a. CPU
 - b. Network
 - c. Disk
 - d. Saturation

* What is AWS Cloud Trail?

AWS CloudTrail increases visibility into your user & resource activity by recording AWS management console actions & API calls. You can identify which users & accounts called AWS, the source IP addresses from which the calls were made, & when the calls occurred.

* Cloud Trail vs CloudWatch ?

CloudWatch monitors performance
→ what amount CPU consumes, RAM consumes etc

CloudTrail monitors API calls in the AWS platform.

→ It monitors instructions given to AWS.

* Steps to create alarm?

First create a server by launching instance
Now, copy the server id.

Now, open CloudWatch by searching in search bar → click on In alarms → Create alarm → select metric → Now → paste copied id in this metrics → select EC2 pre → select CPU utilization by selecting server name → select metric → give name metric name → as-CPU_Utiz → Now → select < 50% → duration select as 2,5,2,5 → Next in alarm → send notification to SNS → EC2 actions → add EC2 action → stop instances → next → Name alarm as - CPU_Utiz_more → next → create alarm,

Next → All alarm → CPU_Utiz_more then we will see as graph.

* Now, how stress can be given to the server by manually as follows, i.e 1st install stress software by using link.

= # apt-get install stress
by entering this link the stress software is gets starts installing.

Now, to add load we use,
= stress -c 5

Now, by this link we added load.
Now to check,
go in alarm → view metric

* Ansible Facts :-

- Ansible facts is a way of getting the "info" of Hosts (nodes)
- This module is automatically called by playbooks to gather useful variables about remote hosts that can be used in playbooks
- It can also be executed directly by `/usr/bin/ansible` to check what variables are available to a host. Ansible provides many facts about the sys, automatically.
- This module is also supported for windows targets
- It's not always required.

\$ ansible all -i myhosts -m setup -a
"filter = * family *"

* I want to install webserver on both ubuntu & redhat by same playbook

* Ansible Modules *

- Ansible with a number of modules
- Modules are discrete units of code that can be used from the command line or in a playbook + task
- Ansible executes each module, usually on the remote target node, & collects return values.
- Each module supports taking arguments. Nearly all modules take key = value arguments, space delimited.
- Some modules take no arguments, & the command / shell modules simply take the string of the command you want to run

yum

Date: 18/1/2023

SUN MON TUE WED THU FRI SAT SUN

apt

service

Get url

etc

And can write own module :

* Custom Inventories *

- static → fix.
- dynamic - changeable.

static inventory

[mywebservers]

public add (If diff. N/w or diff. cloud)

private add (if same N/w)

public DNS name (if diff. n/w or diff. cloud)

private DNS name (if same N/w)

for example. com.

[mydbservers]

one.example.com

two.example.com

Inventories are 2 types.

① Default inventories

② custom inventories

Java is used to run playbook

Date: 18/1/2023

MON TUE WED THU FRI SAT SUN

* Installing Java JDK *

Open master server & copy ssh id & past in bash file i.e run master file.

: sudo -i

: su - uromila

: cd

: ls

: cd myansible_6pm

Now, go in vs code → Create new file
→ Name it → myjavatile.yml → enter

- hosts: all

become: yes

tasks:

- name: I want to install java jdk 8
on ubuntu server

apt:

name: openjdk-8-jdk

state: present

when: ansible_os_family == "Debian"

- name: I want to install java jdk 8
on Redhat server

yum:

name: java-1.8.0-openjdk

state: present

when: ansible_os_family == "RedHat"

On google, Install Open java jdk 8
on ubuntu & Redhat

1st consider On ubuntu

name is ; openjdk-8-jdk,

Date: 18/1/2023

MON TUE WED THU FRI SAT SUN

2nd consider on Redhat
name is java - 1.8.0 - openjdk.
write this names in above
myjavofile.yml playbook
Now save it,

Now in Terminal of vs code,

git add -A

git commit -m "family"

git push.

Noay

In Master bash file

: git pull

enter

ls

enter.

user@user:~/myansible-6pm ansible-playbook
-i myhosts myjavofile.yml

Now, enter,

after executing above statement we
can see that,

on ubuntu,

openjdk-8-jdk — is installed

& on Redhat,

java-1.8.0-openjdk — is installed.

i.e

We can install specific packages
on specific server using families.

* Ansible Variables

① Group ② host

- Variables can define in inventory
- Variables can define & uses in playbook

If we want to install some packages on some nodes instead of all nodes, then we have to make groups for diff nodes, so that we can install packages on particular nodes by making groups.

e.g.

[WebServer group]

[database Server group]

[Nw Server group]

While this,

In inventory code

hosts: write here group name instead of all

then, in terminal

git add -A

git commit -m "group"

git push

Now, In Master bash

umnilo : git pull.

Then run playbook by using foll.

umnilo : ansible-playbook -i myhosts
myjavafile.yml

Now,

In hostsfile → create variables for java

E-myweb

Date: 18/1/20 23

MON TUE WED THU FRI SAT SUN

[mywebserver : Var]

myjavajdk 8 on ubuntu = openjdk-8-jdk

myjavajdk 8 on RedHat = java-1.8.0-openjdk

Now, use this variables defined in hosts file in myjavofile.yml. In myjavofile.yml we should write by following way,
for ubuntu
name: " \${ myjavajdk8 on ubuntu } "

for RedHat

name: " \${ myjavajdk8 on RedHat } "

The use of variable define is,
If want to change Openjdk-8-jdk by Openjdk-9-jdk then we can change only in myhosts only not in myjavofile.yml.

i.e. By changing variable in myhosts we can change packages which to be installed on servers.

Instead of defining variables in inventory or host file, define in group vars folder

Group variable :-

In VS code open new folder by right clicking → Name it as a → group vars

Right click on this folder & create a new file → give the name ^{same} as given to group names in hosts file i.e.

e.g. mywebserver & enters

Now,

In this write something like

myjavajdkonubuntu: openjdk-8-jdk

myjavajdkonRedHat: java-1.8.0-openjdk

& delete this from inventory file.

Then,

git add -A

git commit -m "group vars"

git push

Now,

In Master bash file

curl -sSf https://raw.githubusercontent.com/Ansible/ansible-playbook/master/myplaybook.yml | ansible-playbook -i myhosts myjava.yml

-i myhosts myjava.yml

After executing this the java file is successfully installed on master, node & RedHat servers.

Host Variable :- (Node variable)

In VS code, open new folder by right clicking → Name as → host_vars

Now create new instance & update it.
Now copy private id of this & past it in my hosts

Now Right click on folder & create a new file & the name must be a created new instance private id.

Now open this file & write

myjavajdk8onubuntu : openjdk-9-jdk

In a ansible host variable has higher priority than group variable

Now,

git add -A

git commit -m "hostvars"

git push

In master bash,

git pull

Now,

ansible - playbook -i myhosts myjavofile.yml

ansible - playbook -i myhosts -v myjavofile.yml
Basic details

ansible - playbook -i myhosts --v myjavofile.yml
medium details

ansible - playbook -i myhosts --v myjavofile.yml
full details

Date 23/10/23

MON TUE WED THU FRI SAT SUN

Instead of defining variables in inventory or host file, define in group-vars folder

Group variable:

In VS code open new folder by right clicking → Name it as a → group_vars

Right click on this folder & create a new file → give the name as same given in group names in hosts file i.e. e.g. mywebserver & enter

Now,

In this write link like

myjavajdkonubuntu: openjdk-8-jdk

myjavajdkonRedHat: java-1.8.0-openjdk

& delete this from inventory file.

Then,

git add -A

git commit -m "group_vars"

git push

Now,

In Masterbash file

curl https://myansible 6pm ansible-playbook

-i myhosts myjavafile

After executing this the java file is successfully installed on master, node & RedHat server.

Host Variable :- (Node variable)

In VS code, open new folder by right clicking → Name as → host vars

Now create new instance & update it.
Now copy private id of this & past it in my hosts

Now Right click on folder & create a new file & the name must be a created new instance private id.

Now open this file & write

myjavajdk8onubuntu : openjdk-9-jdk

In a ansible host variable has higher priority than group variable

Now,

git add -A

git commit -m "hostvars"

git push

In master bash,

git pull

Now,

ansible-playbook -i myhosts myjavofile.yml

ansible-playbook -i myhosts -v myjavofile.yml
Basic details

ansible-playbook -i myhosts --v myjavofile.yml
medium details

ansible-playbook -i myhosts --v myjavofile.yml
full details