AWS Resource Creation Using Ansible

Presented By
Amgoth Pravalika

1.Introduction to Ansible

2.project objectives

3.Ansible Script Methodologies

Method-1:Using inline variables

Method-2:Without Variables

Method-3:Using Outline Variables

WHAT IS ANSIBLE

Ansible is an open-source automation tool for configuration management, application deployment, and task automation. It allows IT admins to automate repetitive tasks like system management and infrastructure configuration. Ansible is agentless, meaning it doesn't require extra software on remote machines, using SSH (Linux/Unix)

01 **Cross-Platform Support**

02 **Flexible & Scalable**

03 **No Special Permissions Needed**

04 **Community & Documentation**

05 **Agentless**

06 **Idempotent**

ANSIBLE PLAYBOOK

An Ansible Playbookis a YAML file that defines a set of tasks to be executed on remote systems. It automates processes like system configuration, software installation, and service management.

How to Write an Ansible Playbook

1.Create a YAML file(e.g:ansible.yaml)

2.Define the hosts: Specify which systems the playbook will run on.

3.List tasks: Each task performs a specific action, like installing a package or starting a service.

4.Run the playbook with the ansible-playbook command (e.g.:ansible-playbook playbook.yml

Project Objectives

1.Automate AWS Resource Provisioning

2.Integrate AWS IAM for Secure Access

3.Simplify Configuration Management

4.Speed Up Deployments

5.Create a Scalable Solution

Using Inline Variables

An inline variable in Ansible is a variable that is defined directly within the task or playbook, typically using the syntax. It's a quick way to insert values without creating a separate variable file or inventory file.

Benefits

Quick and simple: You can define variables on the fly within the playbook or task

Localized scope: An inline variable is only available with in the specific task or play where it's defined. It does not affect other tasks or plays in the playbook.

1.Create an IAM User & Download Access and Secret Keys

2.Launch an EC2 Instance

3.Transfer Your PEM File to EC2 Instance :scp -i pinku.pem pinku.pem ec2-user@:/home/ec2-user

4.Set Permissions for the PEM File :sudo chmod 600 pinku.pem

5.Install Ansible and Dependencies

sudo amazon-linux-extras install ansible2 -y
sudo pip3 install boto3
sudo pip3 install botocore
sudo pip3 install boto

6.Edit Ansible Configuration

cd /etc/ansible:Open the hosts file using vim host -un comment the line and at the top add:

[localhost] 127.0.0.1 ansible_host= ansible_python_interpreter=/usr/bin/python3 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/pinku.pem

7.Configure Ansible Permissions : remove # for inventory permissions

8.Create the Ansible Playbook :Inlinevariable.yml

9.Run the Ansible Playbook :ansible-plalybook Inlinevariable.yml

10.Verify in AWS Console :The resources will be created we can see those resources in aws console

```
ansible-2

osts: localhost
ecome: yes
ather_facts: false  #if you want to hide the host key verification option then use gather_facts: false #####

 Inline Variable ###

ars:
  - aws_access_key: "AKIAQ3EGQE4II7GU3HOXG"
  - aws_secret_key: "0gKaifijJzy3vT3sjonF3s5Q13oFAFiPc7yXz0/W"
    title: "AUTH"
  - vpc_name: pinku_vpc
  - igw_name: pinku_igw
  - pubsubnet_name: pinku_pub
  - pvtsubnet_name: pinku_pvt
  - pubroute_table_name: pinku_pub_rt
  - pvtroute_table_name: pinku_pvt_rt
  - security_group_name: pinku_sg
  - vpc_cidr_block: '11.0.0.0/16'
  - pubsubnet_cidr_block: '11.0.1.0/24'
  - pvtsubnet_cidr_block: '11.0.2.0/24'
  - destination_cidr_block: '0.0.0.0/0'
  - port22_cidr_block: '0.0.0.0/0'
  - region: "us-east-1"
  - pubzone: "us-east-1a"
  - pvtzone: "us-east-1b"
  - image_id: "ami-0ca9fb66e076a6e32"
  - type: "t2.micro"
  - instance_name: pinku-ec

asks:

## VPC Creation ####

  - ec2_vpc_net:
      aws_access_key: "{{ aws_access_key }}"
      aws_secret_key: "{{ aws_secret_key }}"
      cidr_block: "{{ vpc_cidr_block }}"
      name: "{{ vpc_name }}"
      region: "{{ region }}"
INSERT --
```

```
        dns_support: yes
        # enable dns hostnames
        dns_hostnames: yes
        tenancy: default
        state: absent  # to delete Vpc then replace absent instead of present
      register: vpc_result

#### Internet gateway Creation ########

    - ec2_vpc_igw:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "vpc-05d54c509ccc702b5"
        region: "{{ region }}"
        state: absent
        tags:
          Name: "{{ igw_name }}"
      register: igw_result

 ## create an vpc pub subnet


    - ec2_vpc_subnet:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "vpc-05d54c509ccc702b5"
        region: "{{ region }}"
        az: "{{ pubzone }}"      # az is the availability zone
        state: absent
        cidr: "{{ pubsubnet_cidr_block }}"
        # enable public ip
        map_public: yes
        resource_tags:
          Name: "{{ pubsubnet_name }}"
      register: pubsubnet_result

    ## create an vpc pvt subnet


    - ec2_vpc_subnet:
-- INSERT --
```

```
    - ec2_vpc_subnet:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "vpc-05d54c509ccc7B2b5"
        region: "{{ region }}"
        az: "{{ pvtzone }}"       # az is the availability zone
        state: absent
        cidr: "{{ pvtsubnet_cidr_block }}"
        # enable public ip
        map_public: no
        resource_tags:
          Name: "{{ pvtsubnet_name }}"
      register: pvtsubnet_result

## create an vpc pub route table


    - ec2_vpc_route_table:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "vpc-05d54c509ccc7B2b5"
        region: "{{ region }}"
        state: absent
        tags:
          Name: "{{ pubroute_table_name }}"
        subnets: "subnet-083Tbdf07fadb957d"
        # create routes
        routes:
          - dest: 0.0.0.0/0
            gateway_id: "{{ igw_result.gateway_id }}"
      register: public_route_table

    ## create an vpc pvt route table


    - ec2_vpc_route_table:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
-- INSERT --
```

ansible-2

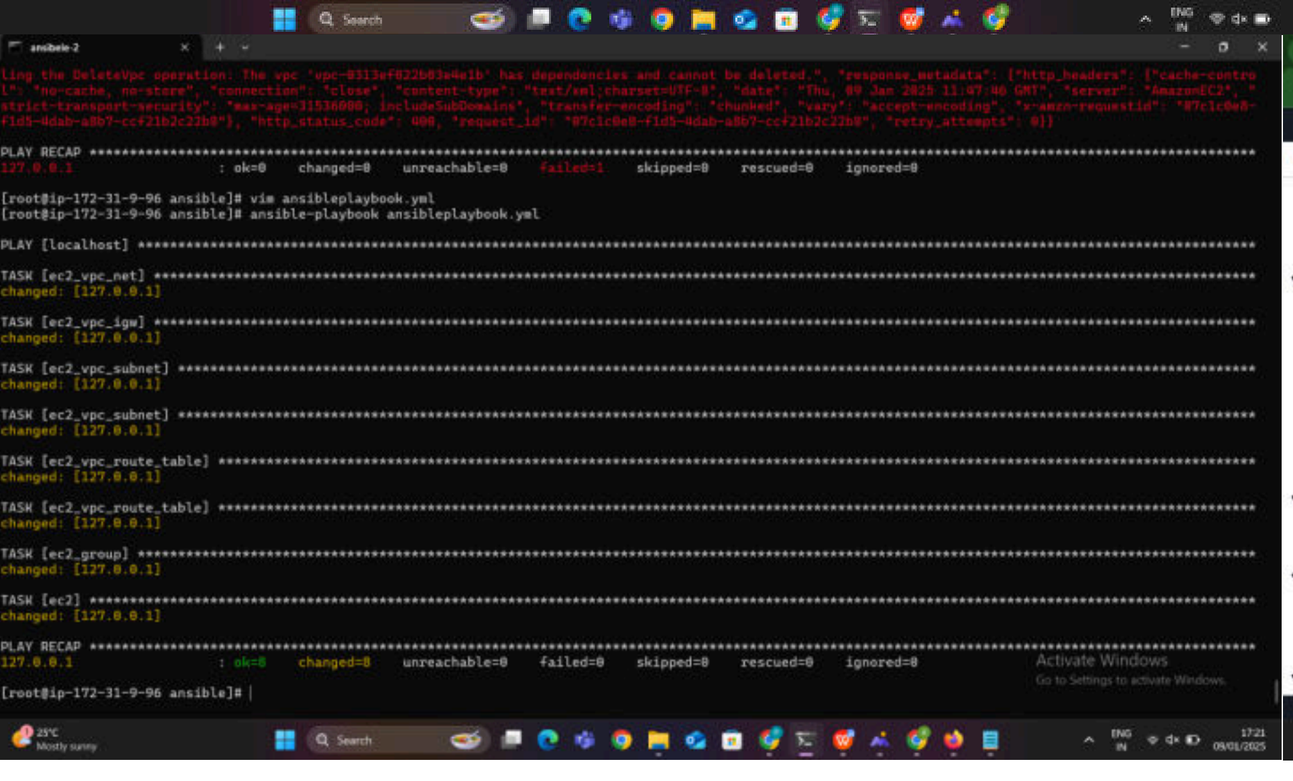
```
      register: public_route_table

    ## create an vpc pvt route table


    - ec2_vpc_route_table:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "vpc-05d54c509ccc7B2b5"
        region: "{{ region }}"
        state: absent
        tags:
          Name: "{{ pvtroute_table_name }}"
        subnets: "subnet-083Tbdf07fadb957d"
      register: private_route_table

    ## create an vpc security groups


    - ec2_group:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "vpc-05d54c509ccc7B2b5"
        region: "{{ region }}"
        state: absent
        name: "{{ security_group_name }}"
        description: allow
        tags:
          Name: Rabbani-sg
        rules:
        - proto: all
          cidr_ip: 0.0.0.0/0
          rule_desc: allow all traffic
      register: security_group_results


    ## tasks file for ec2-launch ##


    - ec2:
-- INSERT --
```

```
 command line flags. ansible will read ANSIBLE_CONFIG,
.cfg in the current working directory, .ansible.cfg in
e directory or /etc/ansible/ansible.cfg, whichever it
irst

]

asic default values...

      = /etc/ansible/hosts
      = /usr/share/my_modules/
tils  = /usr/share/my_module_utils/
mp    = ~/.ansible/tmp
p     = ~/.ansible/tmp
ilters_cfg = /etc/ansible/plugin_filters.yml
      = 5
terval  = 15
r       = root
_pass = True
      = True
t       = smart
ort     = 22
ang     = C
et_locale = False

ill gather facts by default, which contain information about
ote system.

- gather by default, but don't regather if already gathered
it - gather by default, turn off with gather_facts: False
it - do not gather by default, must say gather_facts: True
g = implicit

ly affects the gathering done by a play's gather_facts directive,
ult gathering retrieves all facts subsets
ather all subsets
 --                                                    14,1
```

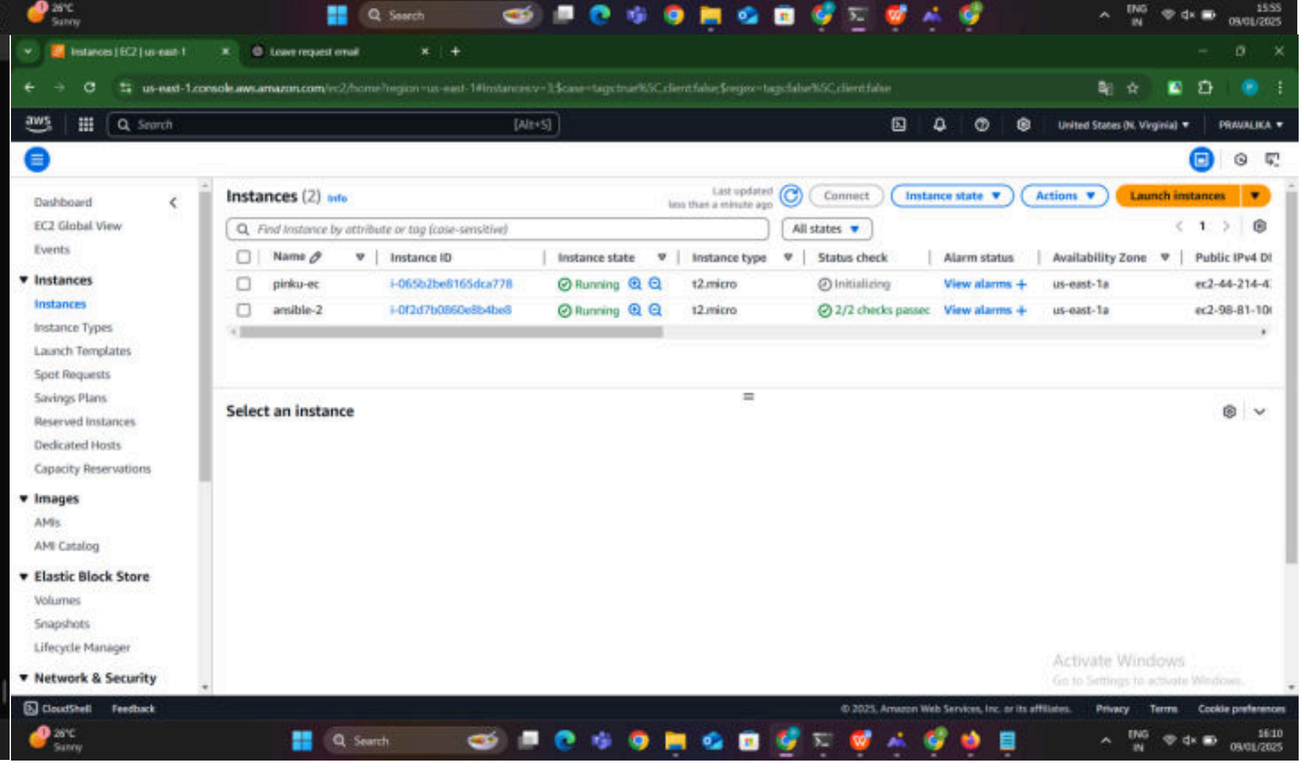
Activate Windows
Go to Settings to activate Windo

```
[localhost]
127.0.0.1 ansible_host=172.31.9.96 ansible_python_interpreter=/usr/bin/python3 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/pinku.pem
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57
-- INSERT --                                          2,155         Top
```

Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.

With Out Variables

An Ansible playbook without variablesis a simple YAML script where all values, such as package names, service names, or configuration settings, are directly specified within the playbook itself. There are no dynamic values or external files used, and every task in the playbook is hardcoded with its parameters.

1.Easy to Write

2.Perfect for Small Tasks

3.No Extra Files

4.Good for One-Time Setup

Internet gateway Creation

```
- ec2_vpc_igw:
    aws_access_key: AKIAQ3EGQE4I7GU34OXG
    aws_secret_key: 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

    vpc_id: "vpc-0042f73cf46aef91a"
    region: us-east-1
    state: absent
    tags:
      Name: pravalika-igw
  register: igw_result
```

create an vpc pub subnet

```
- ec2_vpc_subnet:
    aws_access_key: AKIAQ3EGQE4I7GU34OXG
    aws_secret_key: 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
    vpc_id: "vpc-0042f73cf46aef91a"
    region: us-east-1
    az: us-east-1a      # az is the availability zone
    state: absent
    cidr: 10.0.0.0/20
    # enable public ip
    map_public: yes
    resource_tags:
```

```
- ec2_vpc_route_table:
    aws_access_key: AKIAQ3EGQE4I7GU34OXG
    aws_secret_key: 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

    vpc_id: "vpc-0042f73cf46aef91a"
    region: us-east-1
    state: absent
    tags:
      Name: pravalika-pub-rt
    subnets: [ "subnet-064c291f12328f8ef" ]
    # create routes
    routes:
      - dest: 0.0.0.0/0
        gateway_id: "igw-098e91957fe81f9ac"
  register: public_route_table

## create an vpc pvt route table


- ec2_vpc_route_table:
    aws_access_key: AKIAQ3EGQE4I7GU34OXG
    aws_secret_key: 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
    vpc_id: "vpc-0042f73cf46aef91a"
    region: us-east-1
    state: absent
```

create an vpc security groups

```
- ec2_group:
    aws_access_key: AKIAQ3EGQE4I7GU34OXG
    aws_secret_key: 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
    vpc_id: "vpc-0042f73cf46aef91a"
    region: us-east-1
    state: absent
    name: pravalika-sg
    description: allow
    tags:
      Name: pinku-sg
    rules:
    - proto: all
      cidr_ip: 0.0.0.0/0
      rule_desc: allow all traffic
  register: security_group_results
```

tasks file for ec2-launch

```
- ec2:
    image: ami-0ca9fb66e076a6e32
    instance_type: t2.micro
    instance_id: "i-0d1696144abb3d575"
    region: us-east-1
    wait: yes
    count: 1
    state: absent
    vpc_subnet_id: "subnet-064c291f12328f8ef"
    assign_public_ip: yes
    group_id: "sg-048097d821c73539b"
    aws_access_key: AKIAQ3EGQE4I7GU34OXG
    aws_secret_key: 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
    instance_tags:
            Name: pravalika-Ec
```

Using Outline variables

Outline variablesin Ansible refer to variables that are defined outside the playbook but are used within it. These variables are often stored in separate files (such as inventory files,files, or external sources) and then referenced in the playbook.

1.Reusability

2.Maintainability

3.Flexibility

1.Create ansible playbbok for resources

2.Create secrets.yml playbook

Asecrets.ymlfile in Ansible securely stores sensitive data like passwords and API keys, keeping them separate from your main playbook to enhance security and manageability

VPC Creation

```
- ec2_vpc_net:
    aws_access_key: "{{ aws_access_key }}"
    aws_secret_key: "{{ aws_secret_key }}"
    cidr_block: "{{ vpc_cidr_block }}"
    name: "{{ vpc_name }}"
    region: "{{ region }}"
    # enable dns support
    dns_support: yes
    # enable dns hostnames
    dns_hostnames: yes
    tenancy: default
    state: present  # to delete Vpc then replace absent instead of presnt
  register: vpc_result
```

nternet gateway Creation

```
- ec2_vpc_igw:
    aws_access_key: "{{ aws_access_key }}"
    aws_secret_key: "{{ aws_secret_key }}"
    vpc_id: "{{ vpc_result.vpc.id }}"
    region: "{{ region }}"
    state: present
    tags:
      Name: "{{ igw_name }}"
  register: igw_result
```

```
- ec2_vpc_subnet:
    aws_access_key: "{{ aws_access_key }}"
    aws_secret_key: "{{ aws_secret_key }}"
    vpc_id: "{{ vpc_result.vpc.id }}"
    region: "{{ region }}"
    az: "{{ pvtzone }}"      # az is the availability zone
    state: present
    cidr: "{{ pvtsubnet_cidr_block }}"
    # enable public ip
    map_public: no
    resource_tags:
      Name: "{{ pvtsubnet_name }}"
  register: pvtsubnet_result
```

create an vpc pub route table

```
- ec2_vpc_route_table:
    aws_access_key: "{{ aws_access_key }}"
    aws_secret_key: "{{ aws_secret_key }}"
    vpc_id: "{{ vpc_result.vpc.id }}"
    region: "{{ region }}"
    state: present
    tags:
      Name: "{{ pubroute_table_name }}"
    subnets: [ "{{ pubsubnet_result.subnet.id }}" ]
    # create routes
    routes:
      - dest: 0.0.0.0/0
        gateway_id: "{{ igw_result.gateway_id }}"
  register: public_route_table
```


Thank
you