**A blue and orange text on a white background

Description automatically generated**

# **PG DO - Configuration Management with Ansible and Terraform**

Deploying Web Application Using Ansible

**Git Repository**: https://github.com/sravan1990/

Presented by: Venkata Sravan Kumar Chivukula

.

.

# **Deploying Web Application Using Ansible**

# Project Agenda

To create an automation script to deploy an application using Ansible.

# Scenario

You have joined as a DevOps engineer in XYZ Pvt. Ltd. It is a platform where individuals can create their profile and start blogging on various topics. The application is ready to be hosted on a server. You are tasked with implementing an Ansible script to deploy this application on a remote Nginx server.

# Industry relevance

The following tools used in this project serve specific purposes within the industry:

**1. Ansible:** Ansible automates IT tasks, streamlining configuration management, application deployment, and orchestration. It uses simple, human-readable YAML files called playbooks, allowing easy setup and management of complex IT environments

**Tools required**: Ansible

Terraform,

AWS account with security credentials,

Keypair

**Expected Deliverables**: Launch an EC2 instance using Terraform

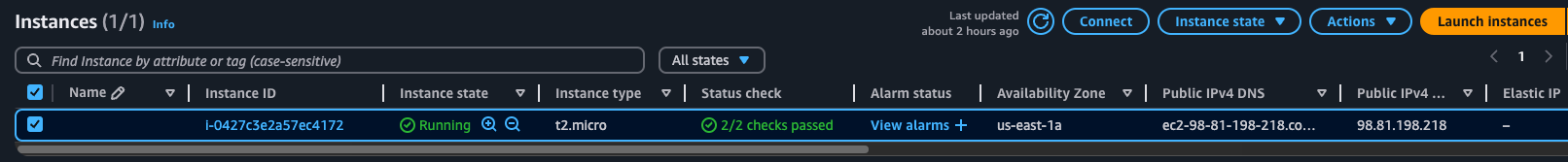
Connect to the instance

Execute the playbook to deploy the web application on the remote server

# Solution Execution Steps

## Launch an EC2 instance using Terraform representing a Remote server

* 1. Create a **main.tf** file. (Check the repository)
  2. initialize terraform using command :   
     *$ terraform init*
  3. Run plan :   
     *$ terraform plan*
  4. run apply :   
     *$ terraform apply*

Now check whether instance is created in AWS.  


## Create an inventory file to define the remote server

### A screenshot of a computer Description automatically generated

## Create a YAML playbook with tasks for installing Nginx, copying web application files, deploying the Nginx configuration, and enabling the site.

* 1. Create a **ngnix.yml** file. (Check the git repository)

## Create a directory for templates and an Ansible playbook for the Nginx configuration

* 1. Templates can be saved under a dedicated directory on the remote  
       
     *var/www/html/application/templates*
  2. Ansible uses a specific type of template called Jinja Templates (.j2)
  3. For the ansible code check the git repository for **ngnix.yml** file.

## Define variables in the playbook for application details and Nginx configuration

* 1. Check the git repository for **ngnix.yml** file.

## Include tasks in the playbook for installing Nginx, copying application files, deploying Nginx configuration, and enabling the Nginx site

* 1. Check the git repository for **ngnix.yml** file.

## Execute the playbook to deploy the web application on the remote server.

* Run playbook  
  *% ansible-playbook -i inventory nginx.yml*A screenshot of a computer

  Description automatically generated

Verify Nginx server installation and restart

A screen shot of a computer program

Description automatically generated

Verify on the browser if the WebServer is online and application runs ☺

A screenshot of a computer

Description automatically generated

## Rerun with better webserver template and modified config

☺ Rerun Playbook with a better website template and a dedicated directory for Templates(Sample Jinja template copied) ☺

A screenshot of a computer

Description automatically generated

The blog site is online ☺ !!!

A screenshot of a computer

Description automatically generated

## Conclusion / Result

Successfully ran an ansible playbook to

* connect to a remote EC2 instance
* Configure and Install Nginx server on the remote
* Copy the templates and website files to remote
* Enable Nginx server
* Update and run application config successfully!!