BOOTCAMP WORKSHOP

By: Shreyansh Sharma, Pawan Yadav, Kushagra Bansal

**Application:** WordPress

**DataBase:** RDS [ MySQL]

**Tool:** Terraform IaC, Jenkins, GitHub

**Cloud:** AWS d/f services

**Features:** Auto Scaling, High Availability, Highly Secure

**Prerequisites:**

Create, GitHub Repository for all team members and make them collaborate such

That each member can contribute.

**Step-1:**

Create user data for each instance to automate the installation for Nginx,

Jenkins, Monitoring, logging, and so on.

<https://github.com/paawanyadav/TTN_Workshop.git>

Here, You can find the required scripts

**Step-2:**

Used Terraform To build the Infrastructure

**VPC:**

We have created a VPC.

Resource used: aws\_vpc

**Subnet:**

Here, We have used 3 subnets

Two public subnets at different availability zones [need for load balancer]

One private subnet for Database RDS.

Resource used: aws\_subnet

**IG & Route Table:**

Here, we create an Internet Gateway to access the traffic

Then using the routing table to map the IG.

Resources name: Aws\_internet\_gateway & aws\_route

**Security Group [SG]:**

Here, we used security group to allow specific ports such as 22,80,8080,3306

We have used “Dynamic Block” for ingress [Reducing the repetition of code]

and one egress.

Also, we have used the DB subnet group to create intercommunication between the instances.

Resources name: aws\_security\_group & aws\_db\_subnet\_group

**RDS:**

Here, we have built the RDS database service to connect to the application.

Public access is off and multi az is true and allow port number 3306.

Resource name: aws\_db\_instance

**Launch Template:**

Here, we used an auto-scaling service and used a template feature to scale up the environment.

In the AWS launch template, we have used user data to automate the configurations

[**https://github.com/paawanyadav/TTN\_Workshop/blob/main/script/nginx\_Script.sh**](https://github.com/paawanyadav/TTN_Workshop/blob/main/script/nginx_Script.sh)

We have created a load balancer and attached the listener and target groups

Resources used: aws\_lb\_target\_group & aws\_lb & aws\_lb\_listener & aws\_launch\_template & aws\_autoscaling\_group

**Instances:**

Here, we have created a Jenkins instance to build an artifact of the project whenever some push to the GitHub using Github webhooks concept

Resources used: aws\_instance

**S3 backend:**

S3 bucket on AWS to store the state and tfstate backup file