Thoughworks Hacker Ranker Test:

Agenda: Launch the EC2 instances. Configure Nginx Webserver in one server and configure Apache Tomcat in another server. Nginx and Apache Tomcat should be enabled with Self-Signed SSL Certificates. Nginx to Apache Tomcat integration. Finally, Nginx will have static content (static.zip) Apache Tomcat should have dynamic content (comanyNews.war).

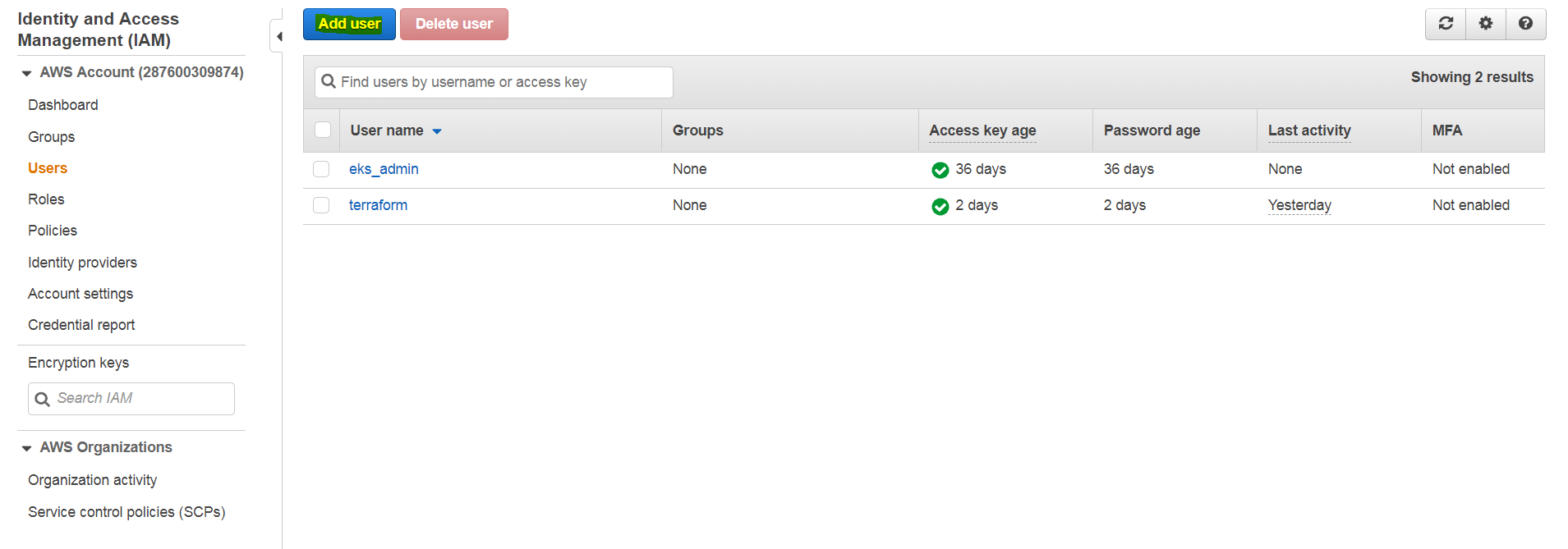
Tools used:

|  |  |
| --- | --- |
| Cloud Provider | Amazon Web Services |
| Terraform | To launch the EC2 Machines |
| S3 | To store the static.zip and companyNews.war files |
| Ansible | Installing and Configuring Nginx and Tomcat servers. Also, for Deploying the applications. |
| Web Server | Nginx |
| Application Server | Apache Tomcat |
| OpenSSL | For generating Self-Signed Certificates |
| IAM User, IAM Policy and IAM Role with EC2FullAccess | To communicate the Server with AWS Account and launch the EC2 machines |
| AWS CLI | For configuring the secure credentials for Terraform access |

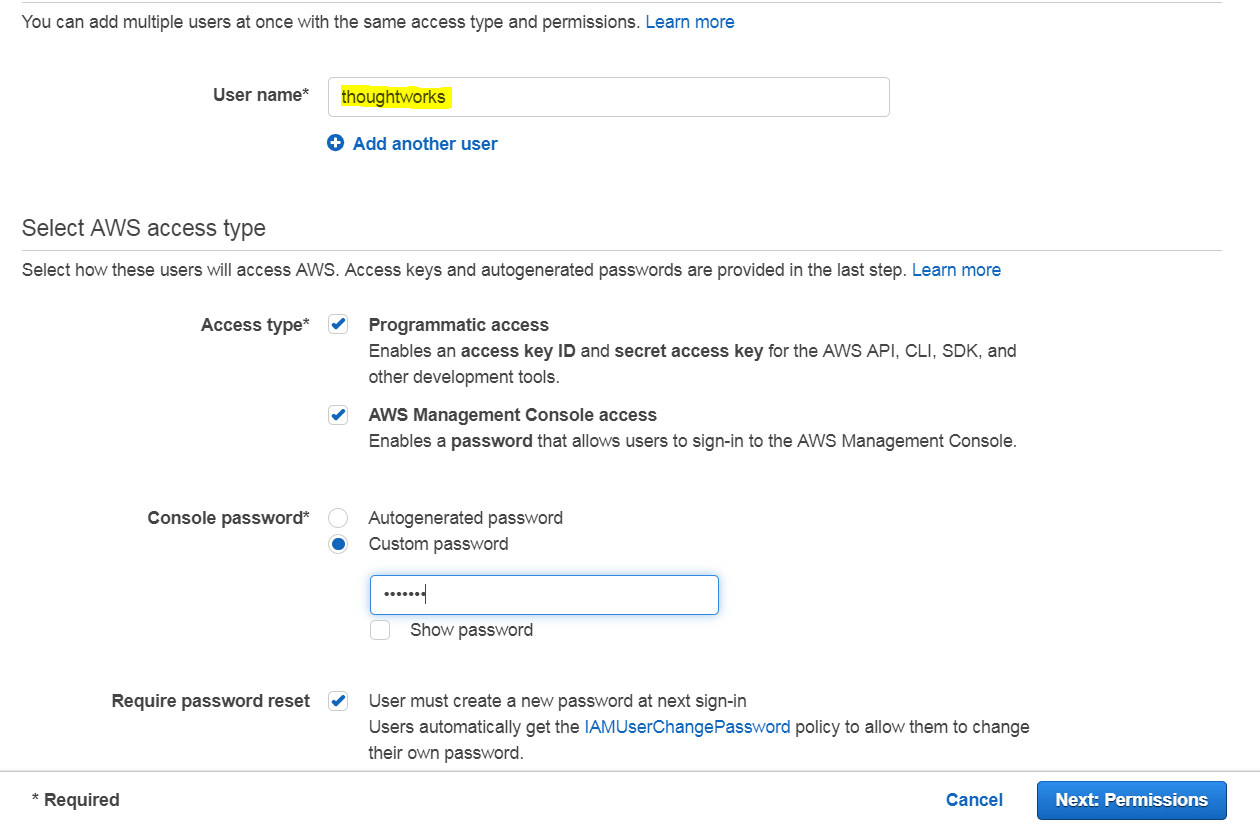
**High-Level Over-view:**

Wrote a terraform script to launch 2 EC2 machines.

Before getting started with Terraform, we need to create an IAM user in IAM section AWS console, Create an IAM role and grant the EC2FullAccess policy to the IAM role. Finally attach the IAM role to the server where Terraform is executing.

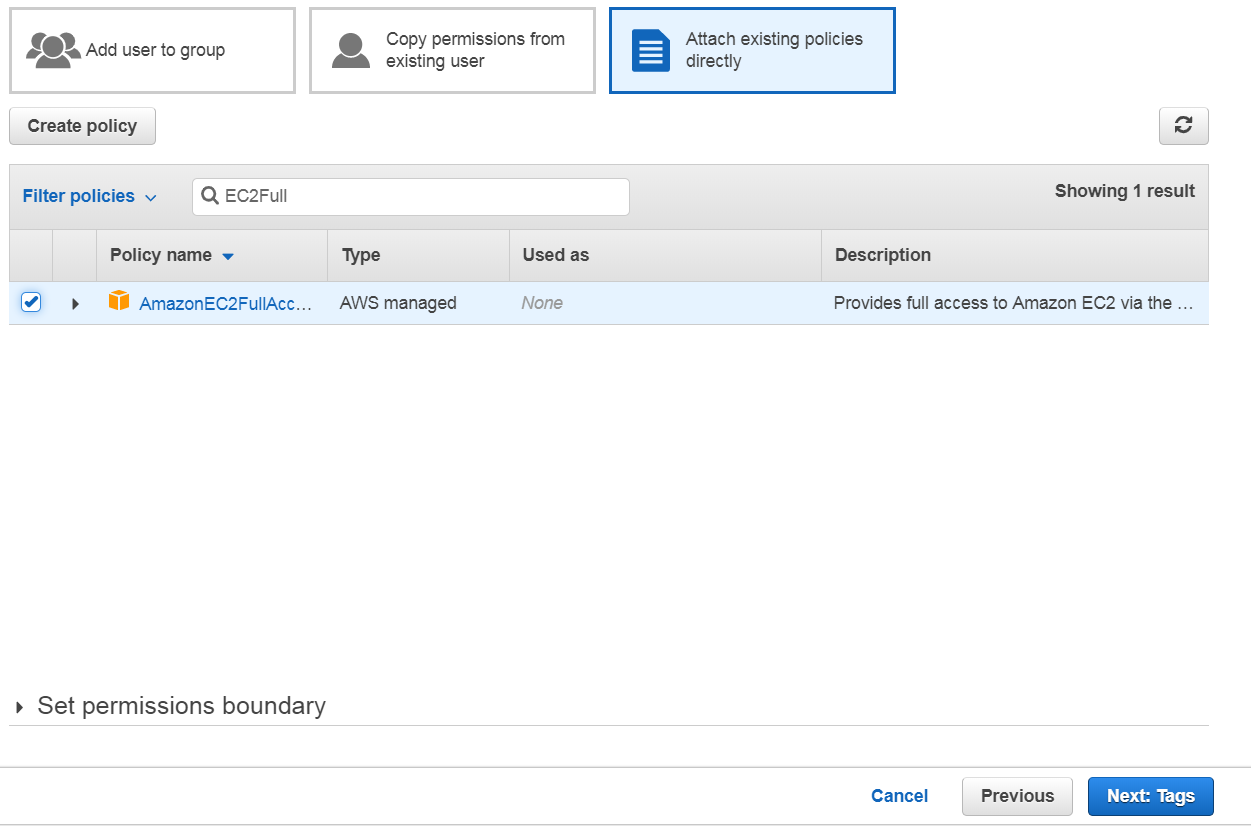


Click on Add User:



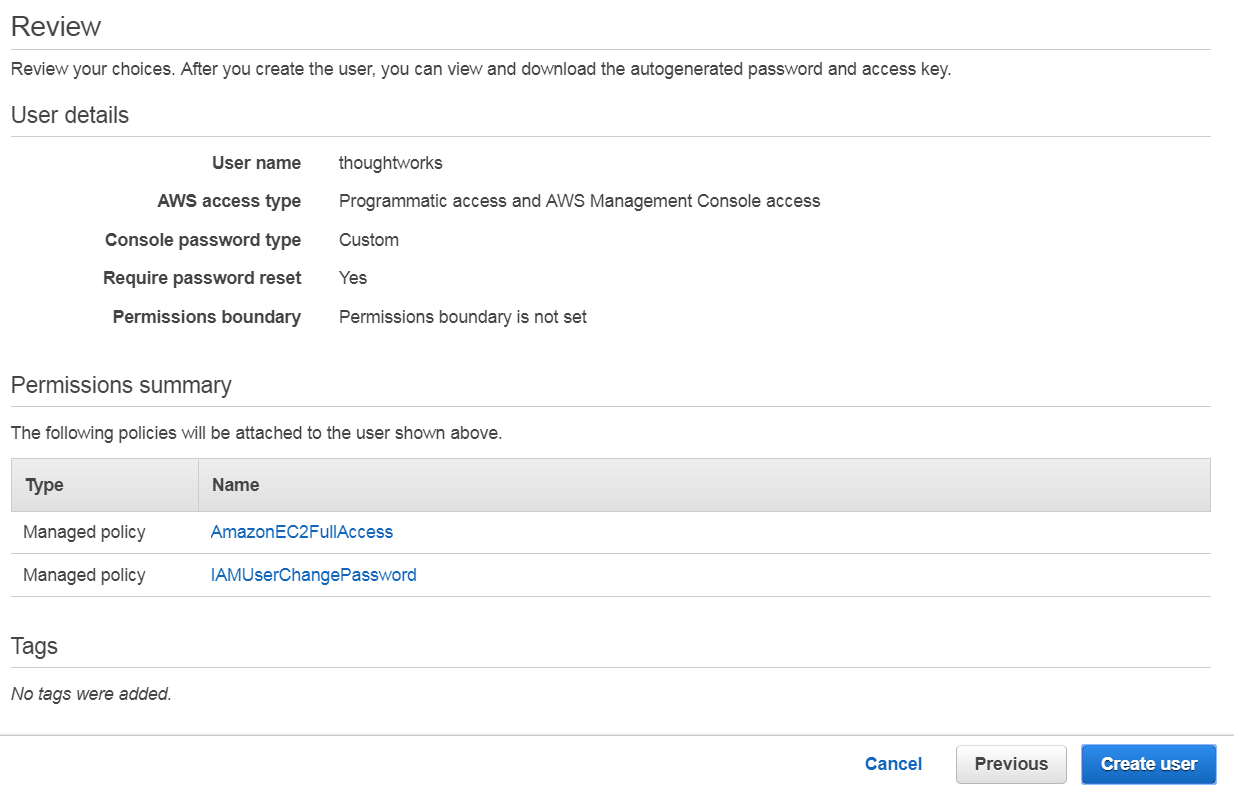
Enter the User name, select the type of access that we want to grant to the user

Create a password for the user. And Whenever user logs-in the user should change the password.

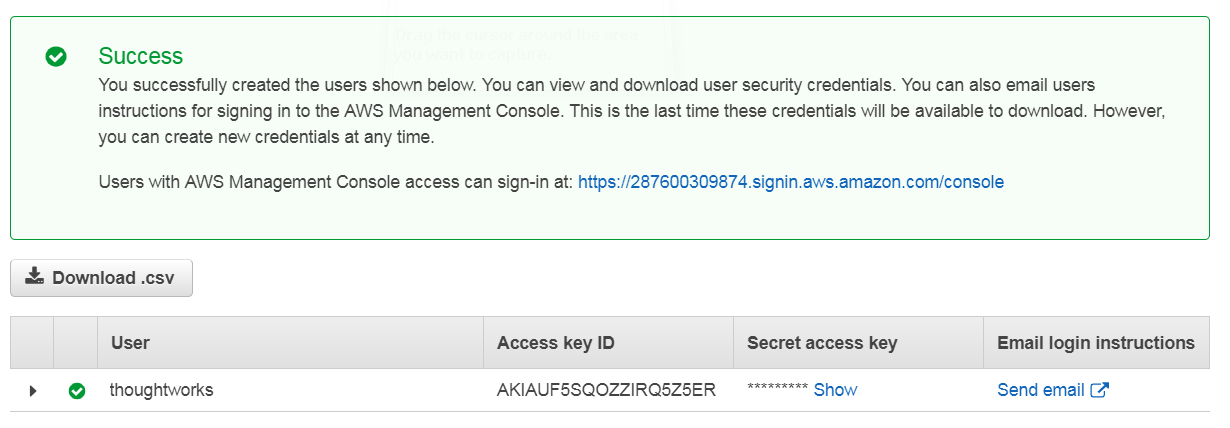


Attach an existing policy directly. Now the new user will have complete access to the EC2 services.

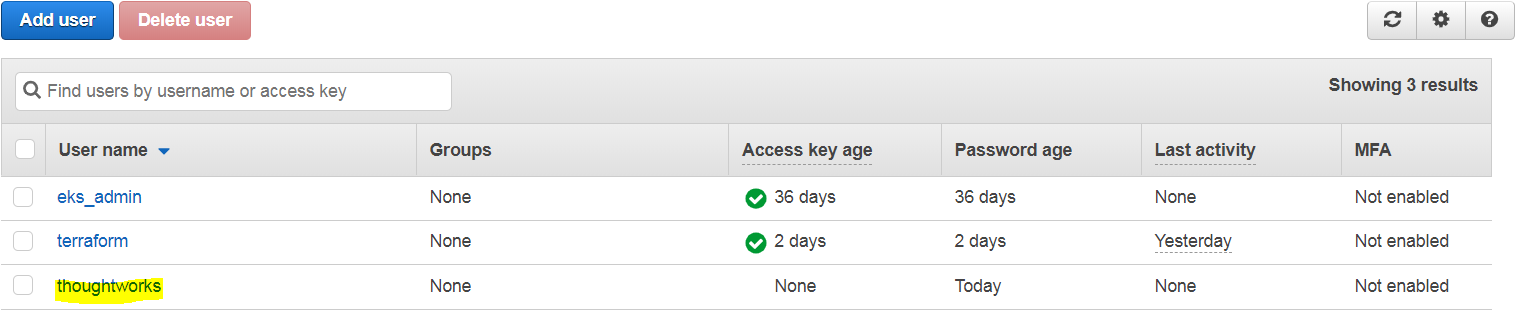
Review and create user.



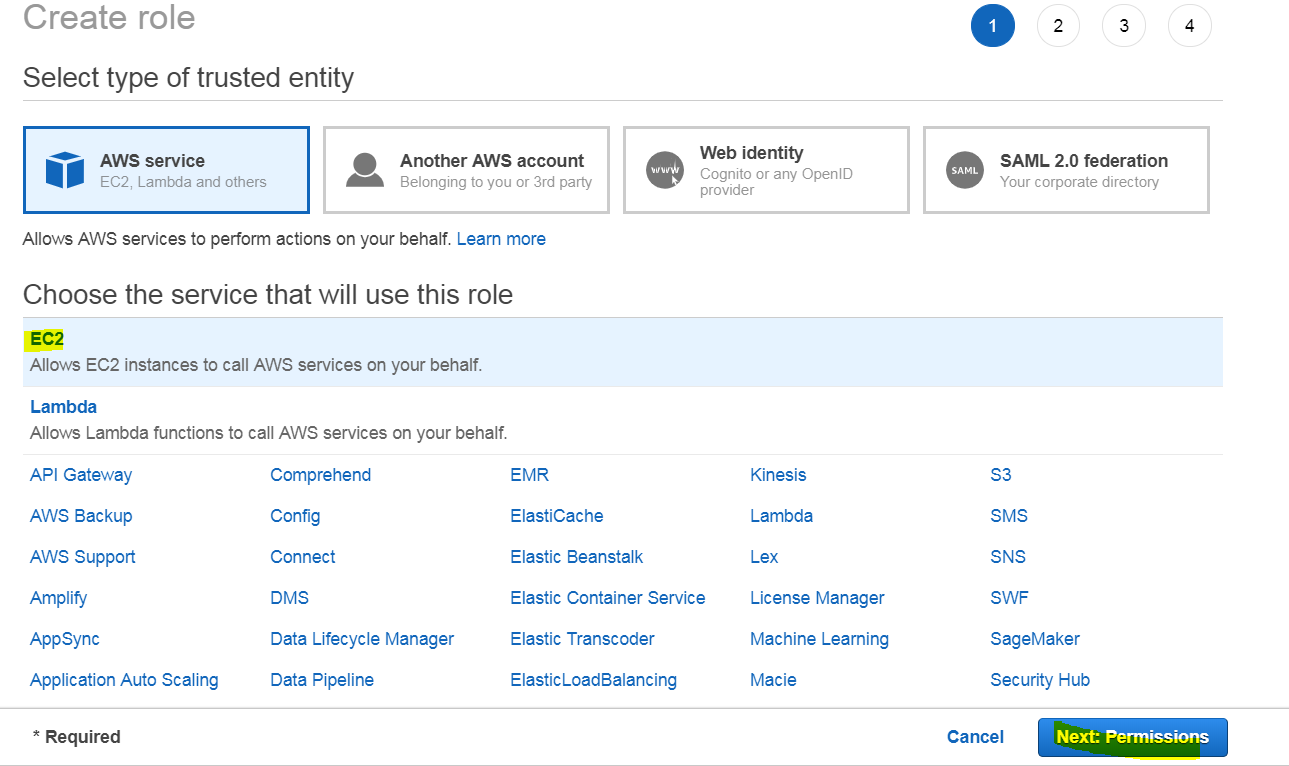
Once we create a user, we will get the Access Key and Secret keys. These keys will be used for configuring the EC2 machine for running the Terraform scripts.

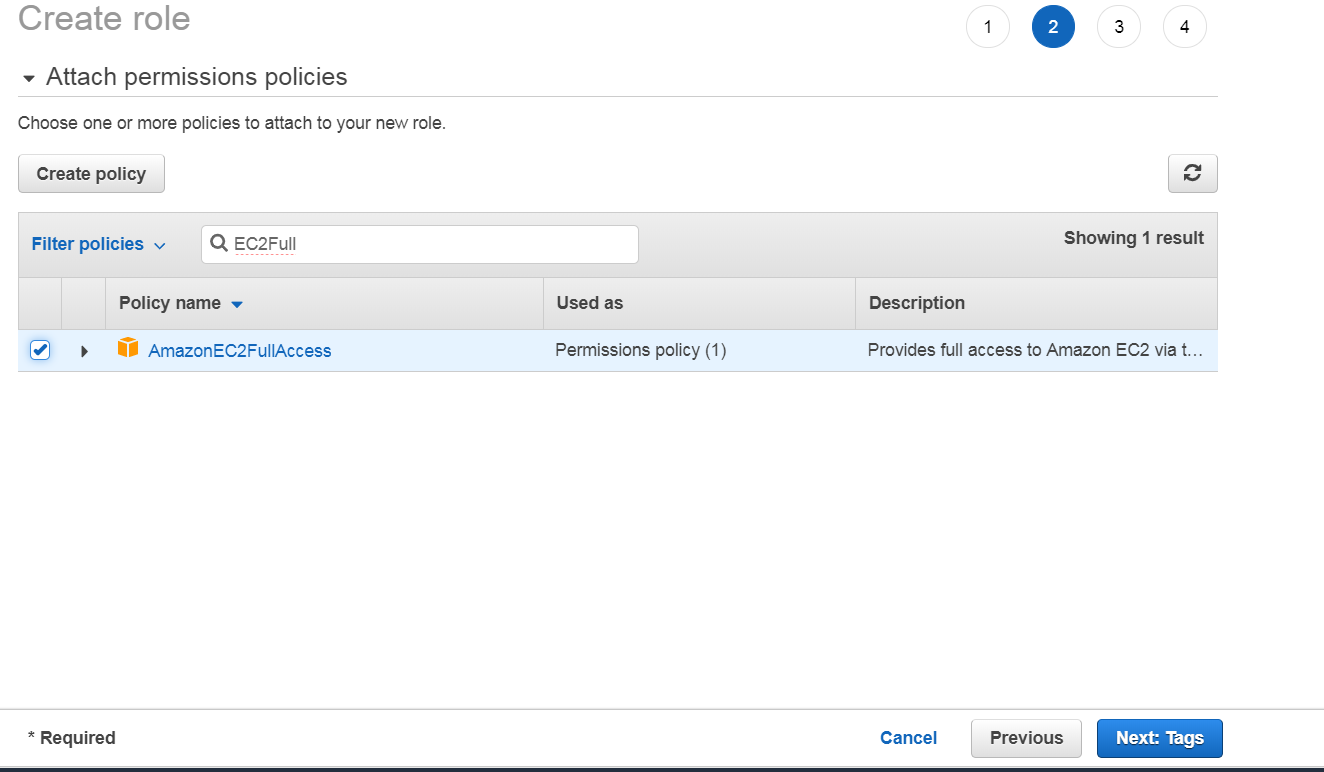


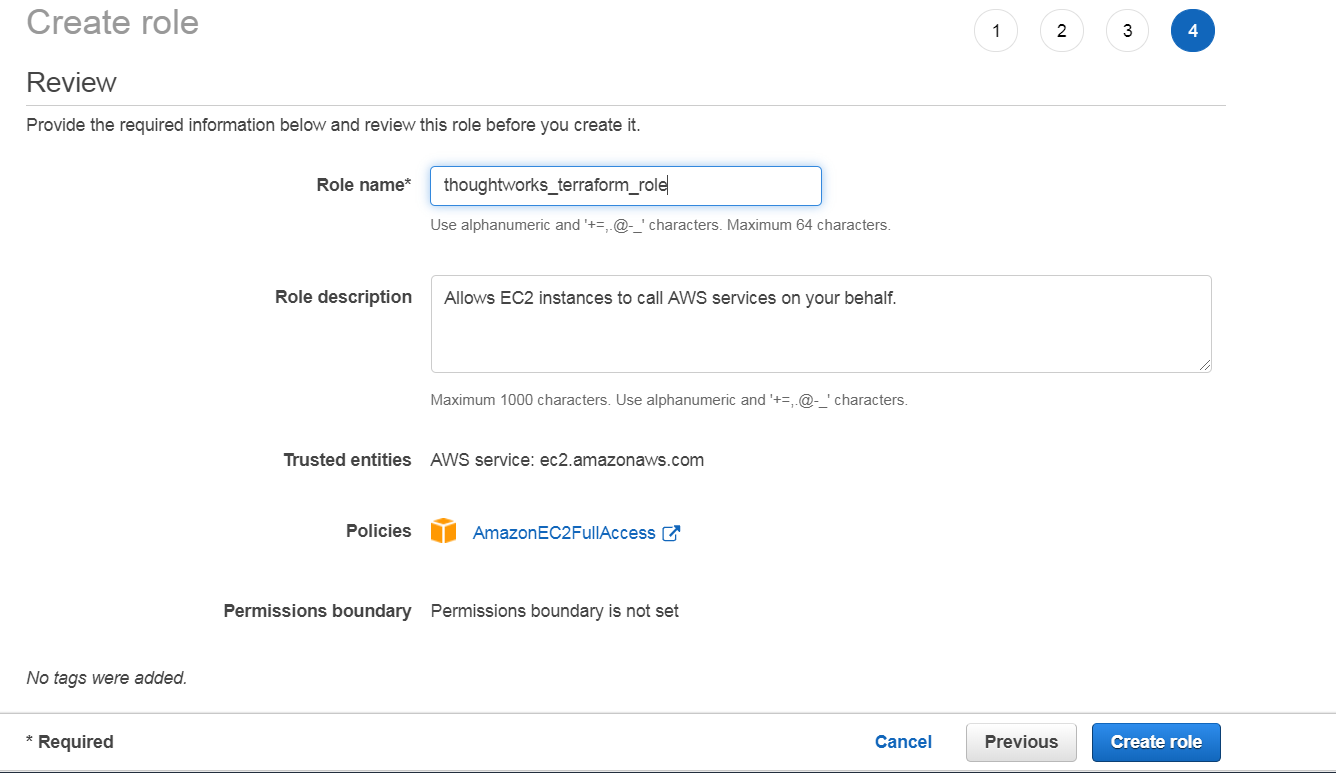
Now the User is ready to use.

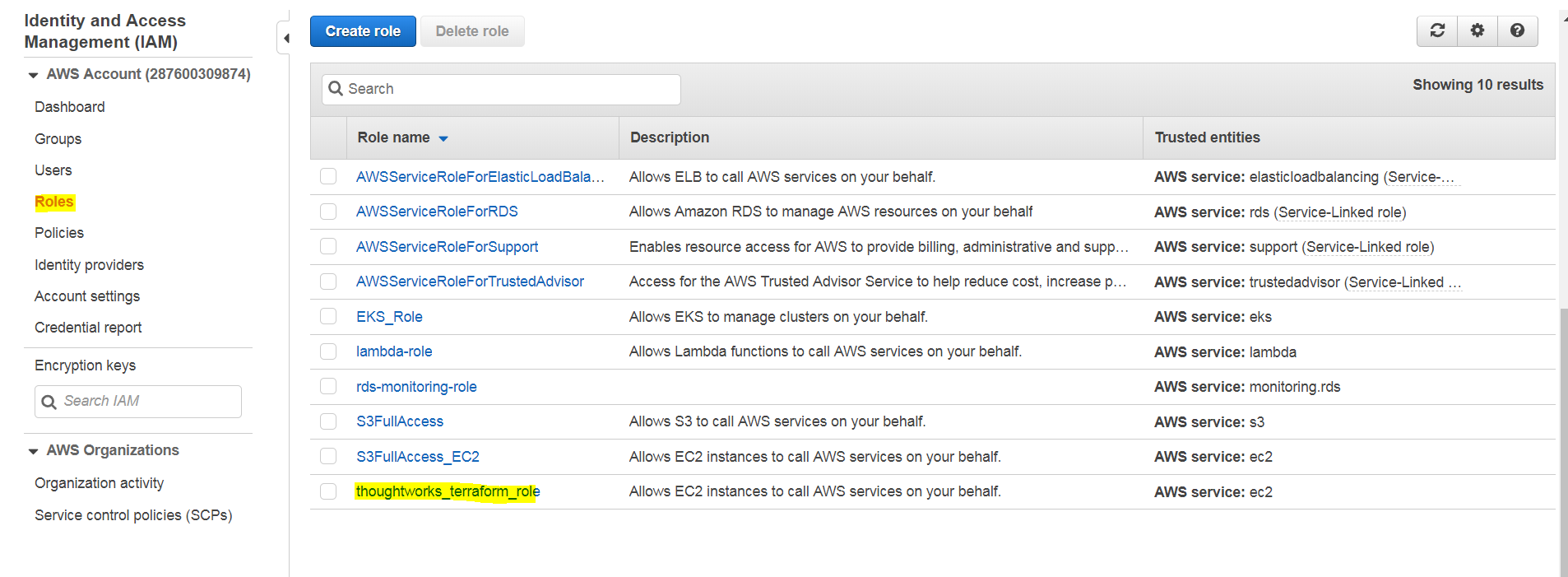


Also create an IAM Role as follows.

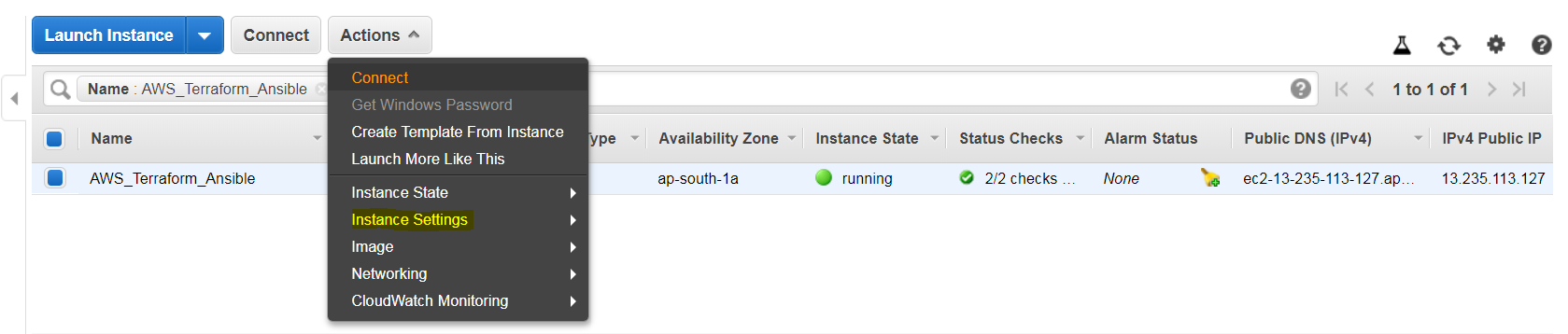




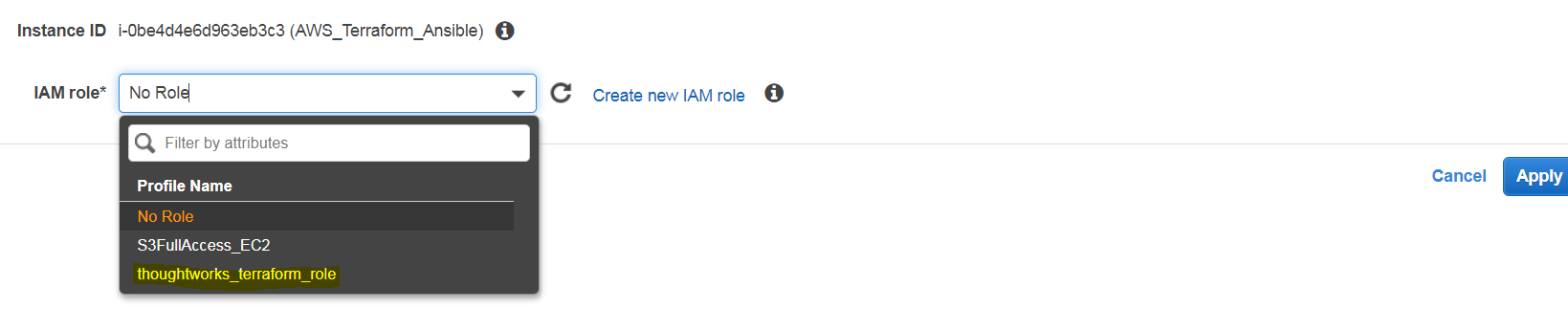


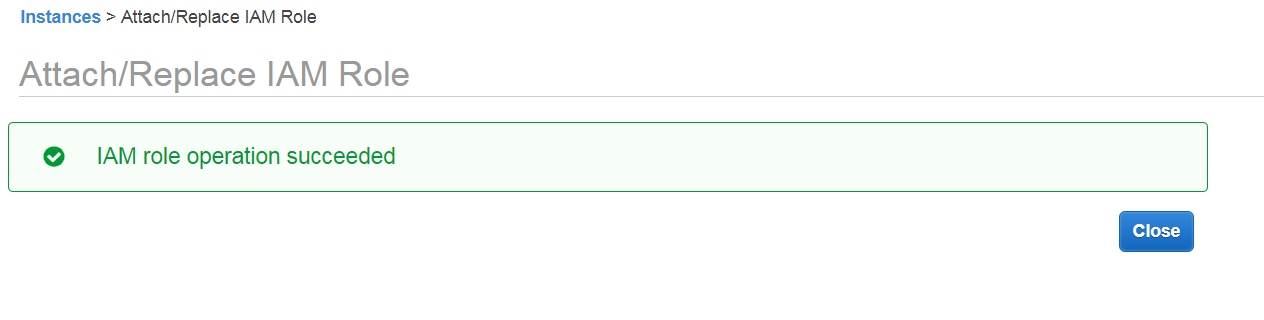


Now the role is available. Attach this role to the EC2 machine where we are running the Terraform Scripts.



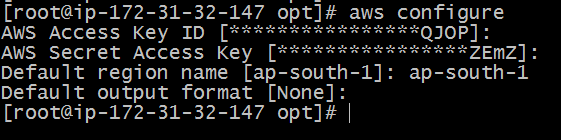
Select the Instance 🡪 Actions 🡪 Instance Settings 🡪 Attach/Replace IAM Role 🡪 Select the role we want to attach (here thoughworks\_terraform\_role) 🡪 Apply





Now we need to configure the AWS access key and secrete keys to the AWS EC2 machine for Terraform.

<https://docs.aws.amazon.com/cli/latest/userguide/install-bundle.html> use this link to install and configure AWS CLI

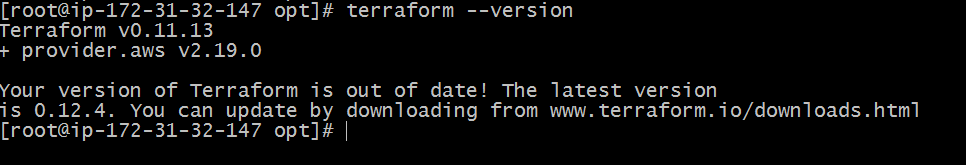


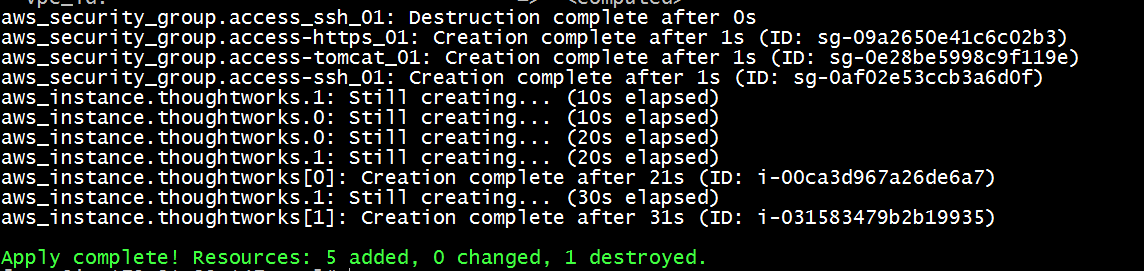
Download and install Terraform for Linux platform.

<https://www.vasos-koupparis.com/terraform-getting-started-install/>

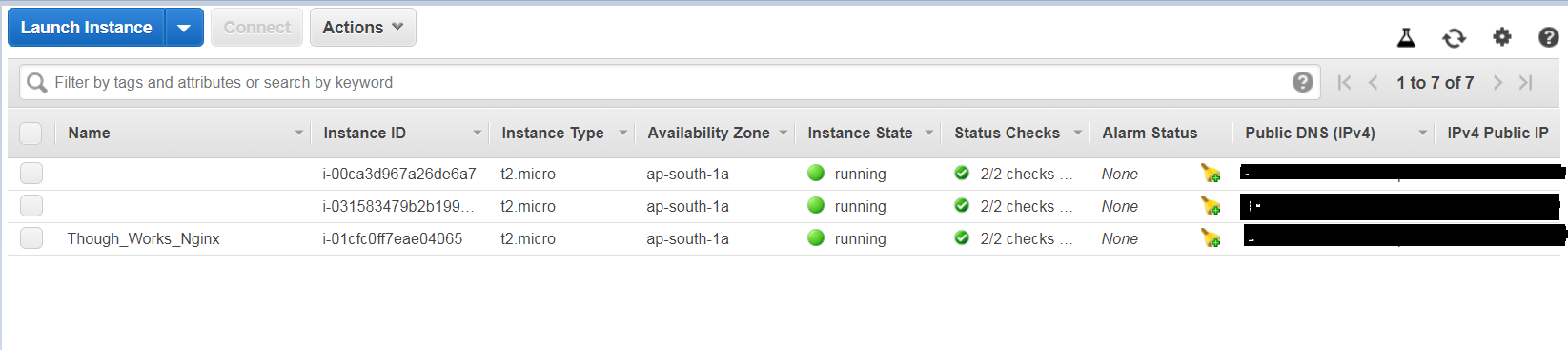
**Install Terraform – Linux** section

Verify Terraform version:

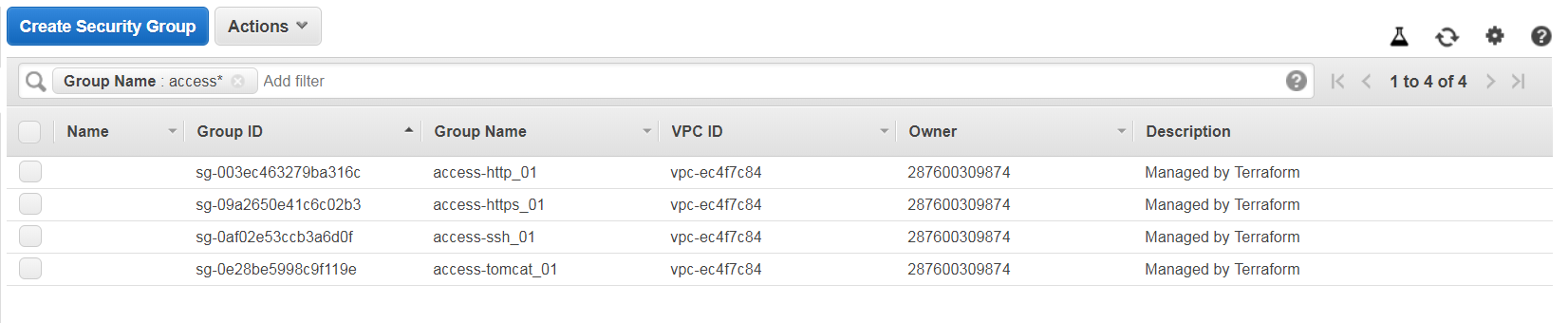




Creating Instances:



Creating Security Groups:



Terraform Script is pushed to githut repository

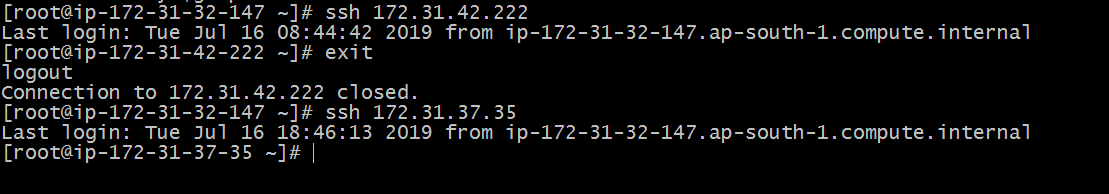
Now the instances are ready.

Now we need to Install and configure the Nginx and Apache Tomcat servers using Ansible.

Follow the below link for Ansible installation on Linux machine

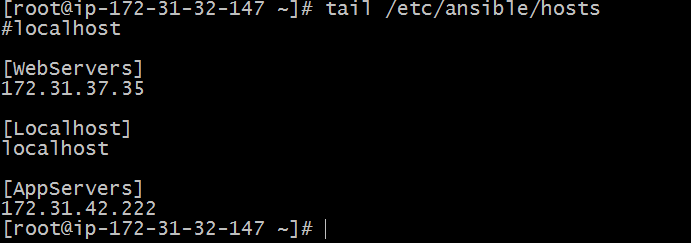
<https://developers.redhat.com/blog/2016/09/02/how-to-install-and-configure-ansible-on-rhel/>

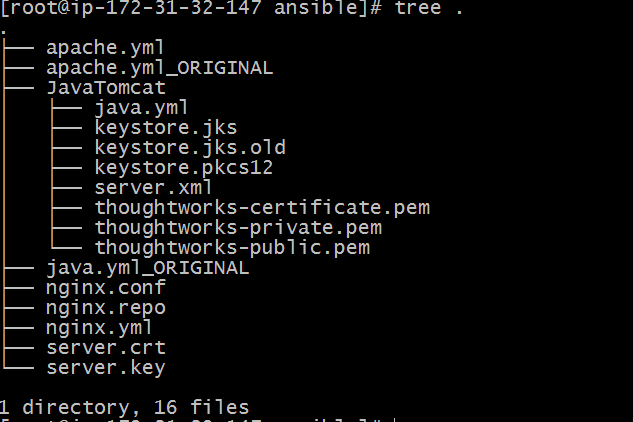
Configure the ssh password less setup from Ansible central server to the remote linux machines using ssh-keygen and copy the id\_rsa.pub file to the two machines and make sure that the password less ssh is working fine.



Inside the server, we can use private IP’s. Here password less ssh is working fine.

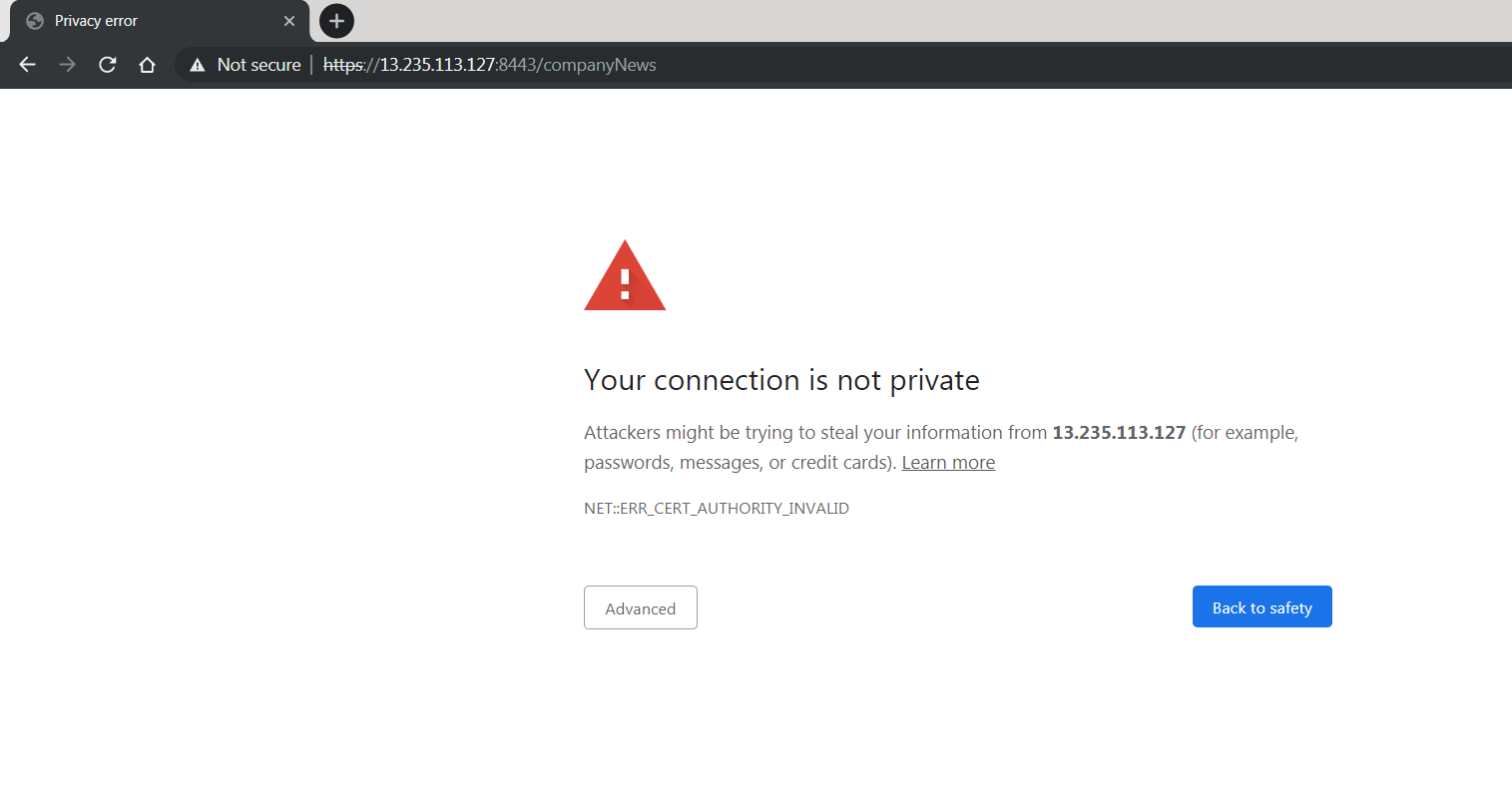
Add the remote server entries into the host inventory file (/etc/ansible/hosts) as follow.

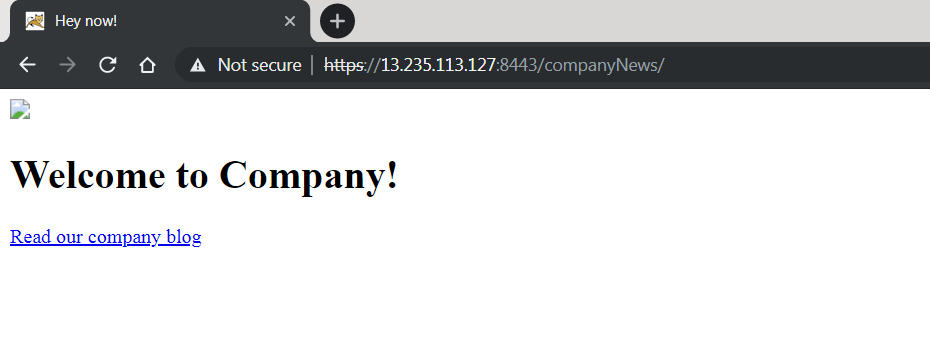


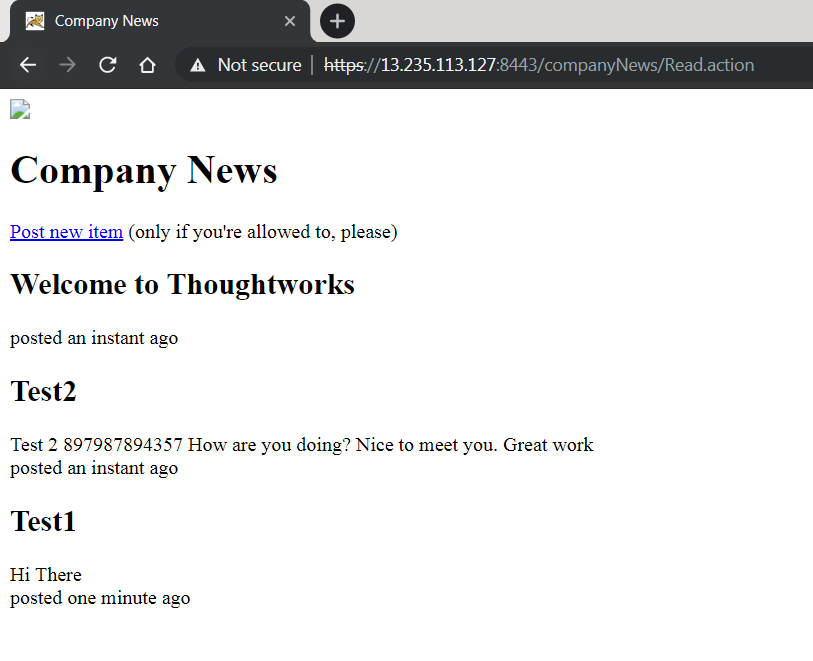


All the ansible yml files, ssl certificates, repo file for nginx installation and custom server.xml files.

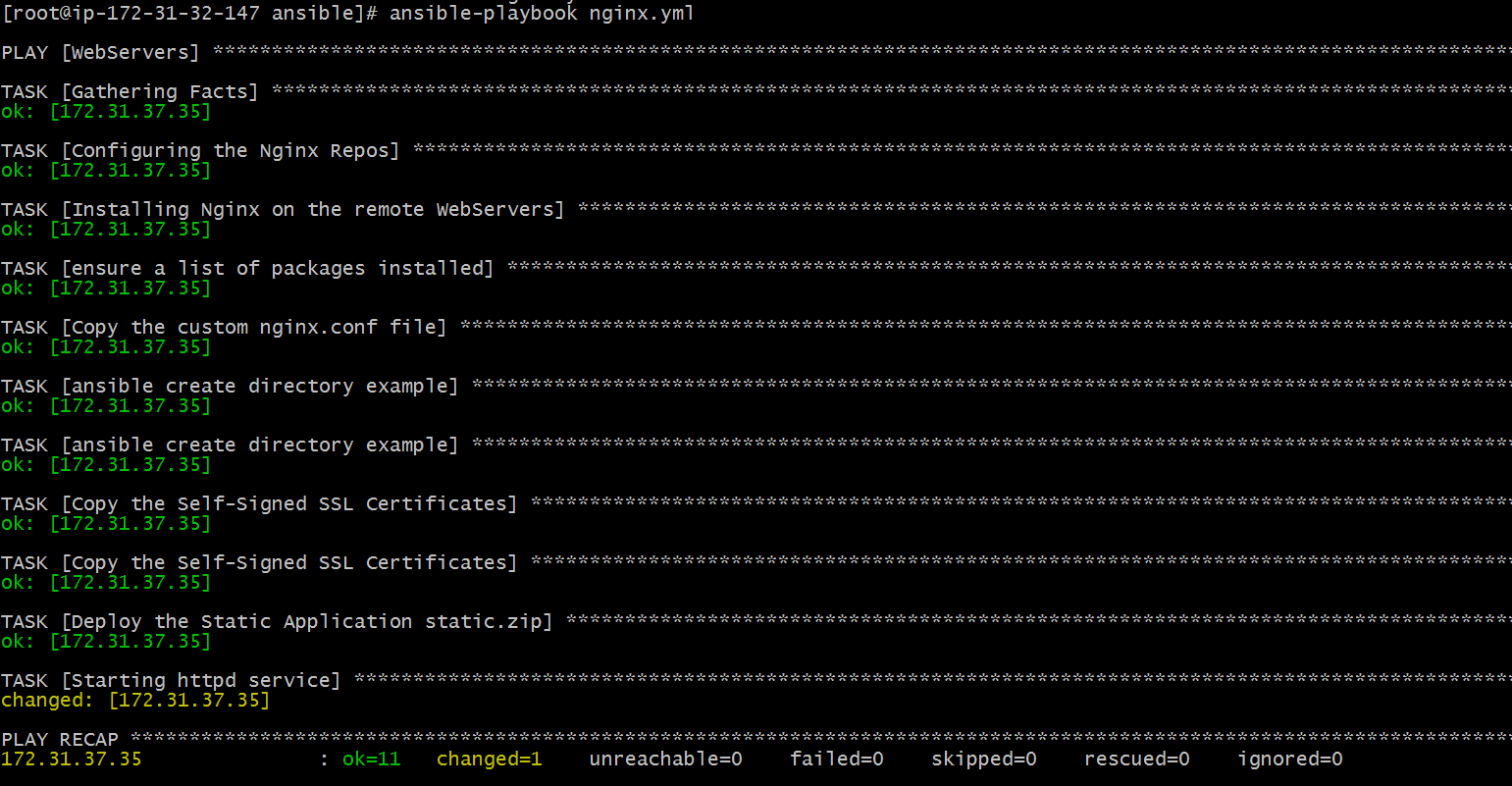
Directly from Tomcat on 8443 port.







**From Nginx SSL Enabled:**



Access the URL:

<https://ec2-13-233-232-186.ap-south-1.compute.amazonaws.com/companyNews>

https://<NginxIP>/comanyNews