

# Installation of ansible tower on AWS

## What is Ansible Tower ?

Ansible Tower (formerly 'AWX') is a web-based solution that makes Ansible even more easy to use for IT teams of all kinds. It's designed to be the hub for all of your automation tasks.

- Ansible tower is ansible at a more enterprise level

### Prerequisites to install ansible tower

The following operating systems support Ansible Tower

RedHat Enterprise Linux 6 64-bit

RedHat Enterprise Linux 7 64-bit

CentOS 6 64-bit

CentOS 7 64-bit

Ubuntu 12.04 LTS 64-bit

Ubuntu 14.04 LTS 64-bit

Ubuntu 16.04 LTS 64 bit

You should have the latest stable release of Ansible.

It required a 64-bit support kernel, runtime, and 20 GB hard disk.

Minimum 2 GB RAM (4 GB RAM recommended) is required.

Minimum 2 GB RAM is recommended for Vagrant trial installations

And 4 GB RAM is recommended /100 forks

AWS need the m4.large instance type

### Step-1

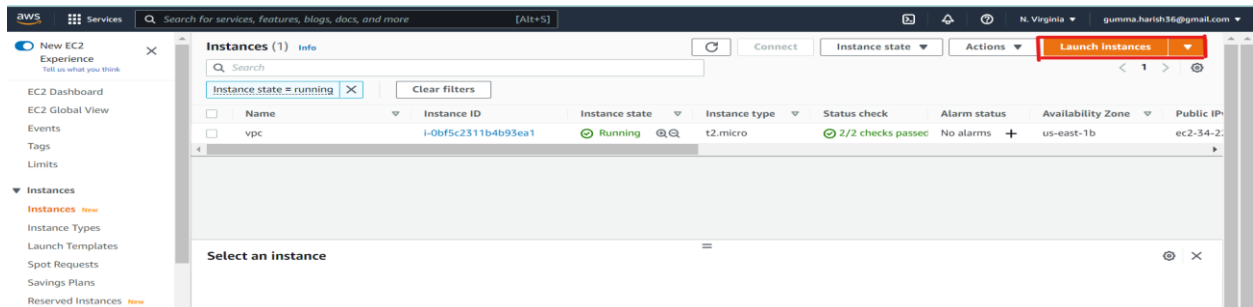
Update system and add EPEL repository

1.update package manager

```
$ sudo yum -y update
```

Install EPEL Repository on RHEL

### Step-2 : Launch the instance




## Choose an Amazon machine image


1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

 **macOS Big Sur 11.6.5** - ami-0dddc64bc72740146  
The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

 **Red Hat Enterprise Linux 8 with High Availability** - ami-0f095f89ae15be883  
Red Hat Enterprise Linux version 8 with High Availability (HVM), EBS General Purpose (SSD) Volume Type  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select 64-bit (Mac)

Select 64-bit (x86)

## Choose an instance type

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: m4 Current generation Show/Hide Columns

Currently selected: m4 large (- ECUs, 2 vCPUs, 2.4 GHz, -, 8 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input checked="" type="checkbox"/>	m4	m4 large	2	8	EBS only	Yes	Moderate	Yes
<input type="checkbox"/>	m4	m4.xlarge	4	16	EBS only	Yes	High	Yes
<input type="checkbox"/>	m4	m4.2xlarge	8	32	EBS only	Yes	High	Yes
<input type="checkbox"/>	m4	m4.4xlarge	16	64	EBS only	Yes	High	Yes
<input type="checkbox"/>	m4	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit	Yes
<input type="checkbox"/>	m4	m4.16xlarge	64	256	EBS only	Yes	25 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

## Configure instance details

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances 1 Launch into Auto Scaling Group

Purchasing option ☐ Request Spot instances

Network vpc-09f7ab90b450f1906 (default) Create new VPC

Subnet No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP Use subnet setting (Enable)

Hostname type Use subnet setting (IP name)

DNS Hostname ☒ Enable IP name IPv4 (A record) DNS requests  
☒ Enable resource-based IPv4 (A record) DNS requests  
☐ Enable resource-based IPv6 (AAAA record) DNS requests

Placement group ☐ Add instance to placement group

Capacity Reservation Open

Domain join directory No directory Create new directory

IAM role None Create new IAM role

Cancel Previous Review and Launch Next: Add Storage

## Add Storage (for ansible tower we need the minimum requirement 20GB .but take little bit extra GB)

AWS Services Search for services, features, blogs, docs, and more [Alt+F5] N. Virginia gumma.harish36@gmail.com

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-06e7cbd0bd9e9885b	40	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Shared file systems  
You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

Add file system

Cancel Previous Review and Launch Next: Add Tags

## Configure security Group (create a new security group)

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group  
☐ Select an existing security group

Security group name:   
Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0 ::/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0 ::/0	e.g. SSH for Admin Desktop

Add Rule

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

## Review Instance launch

### Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Improve your instances' security.** Your security group, **ansibletower**, is open to the world.  
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

**Your instance configuration is not eligible for the free usage tier**  
To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about [free usage tier](#) eligibility and usage restrictions. [Don't show me this again](#)

#### AMI Details

**Red Hat Enterprise Linux 8 with High Availability - ami-0f05f09ae15be883**  
Red Hat Enterprise Linux version 8 with High Availability (HVM), EBS General Purpose (SSD) Volume Type  
Root Device Type: xbs Virtualization type: hvm

[Edit AMI](#)

#### Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
m4.large	-	2	8	EBS only	Yes	Moderate

[Edit instance type](#)

#### Security Groups

[Edit security groups](#)

[Cancel](#) [Previous](#) [Launch](#)

## Create a new key pair and download the key pair and launch the instance

### Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair  
**Key pair type**  
☒ RSA ☐ ED25519  
**Key pair name**  
  
[Download Key Pair](#)

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

[Cancel](#) [Launch Instances](#)

## Launch status

### Launch Status

**Get notified of estimated charges**  
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

#### How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

#### Here are some helpful resources to get you started

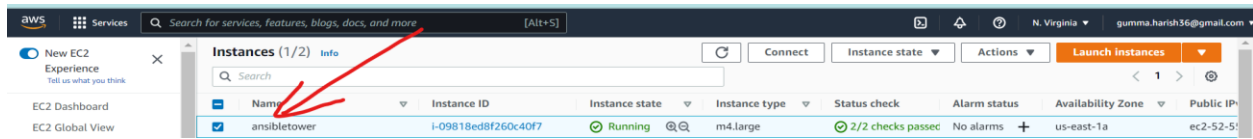
- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes. (Additional charges may apply)
- Manage security groups

[View instances](#)

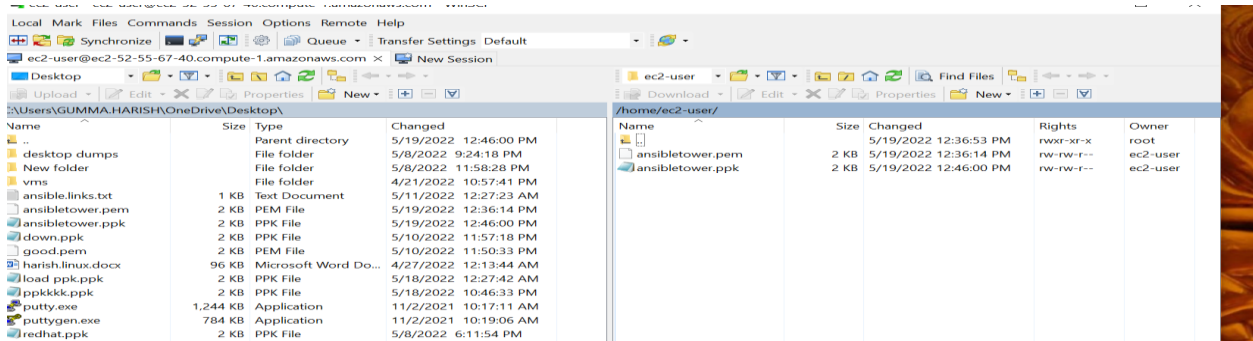
Now instance has created



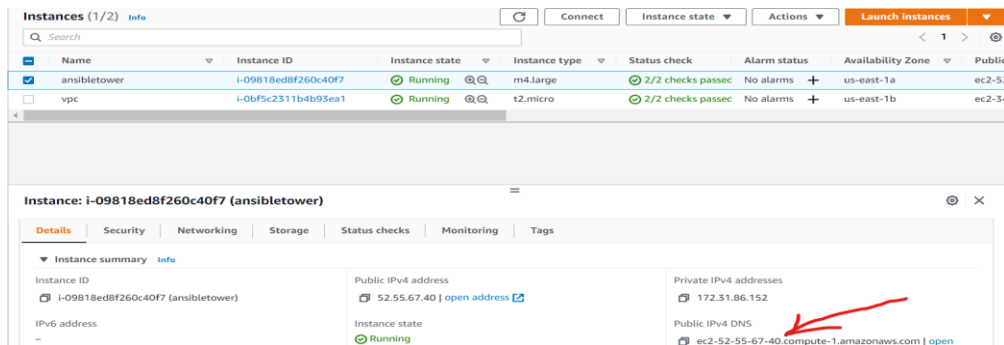
Pem file and should be available in the server (means it should be in linux machine)

Using the winscp are mobaxterm for transferring the files from windows to linux

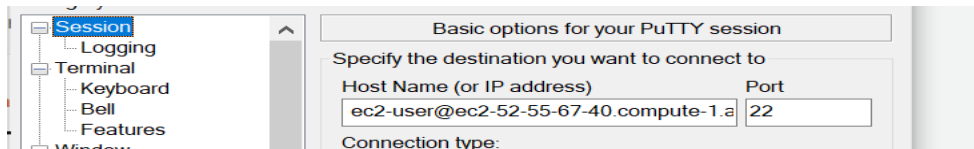
Winscp is used to transfer the file from windows to linux



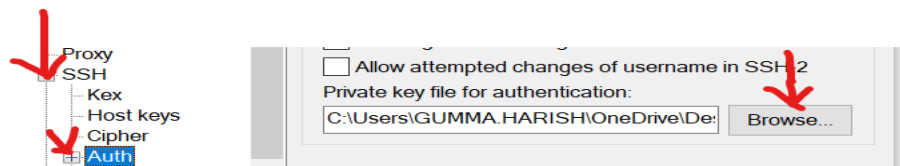
Now using the putty we connect we need to take the aws Dns name



Putty connection



Add the ssh under auth we need to browse



### Step-3

Update the package manager

**\$ sudo yum update -y**

```
yum 1:7.0-1.el8.noarch
zlib-1.2.11-18.el8_5.x86_64
Installed:
NetworkManager-initscripts-updown-1:1.36.0-4.el8.noarch      glibc-gconv-extra-2.28-189.1.el8.x86_64      grub2-tools-efi-1:2.02-123.el8.x86_64
kernel-4.18.0-372.9.1.el8.x86_64                          kernel-core-4.18.0-372.9.1.el8.x86_64        kernel-modules-4.18.0-372.9.1.el8.x86_64
python3-netifaces-0.10.6-4.el8.x86_64
Complete!
```

Install EPEL Repository on RHEL

**\$ sudo dnf install**

<https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm>

```
Complete!
[ec2-user@ip-172-31-86-152 ~]$ sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
Updating Subscription Management repositories.
```

Output

```
Verifying : epel-release-8-15.el8
Installed products updated.

Installed:
epel-release-8-15.el8.noarch
Complete!
```

**\$ sudo dnf repolist epel**

```
[ec2-user@ip-172-31-86-152 ~]$ sudo dnf repolist epel
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

repo id      repo name      status
epel         Extra Packages for Enterprise Linux 8 - x86_64    enabled
```

Ansible tower uses ansible playbook to deploy itself so we also need ansible installed

**\$ sudo yum install ansible vim curl -y**

```
[ec2-user@ip-172-31-86-152 ~]$ sudo yum install ansible vim curl
Updating Subscription Management repositories.
```

Output

```
Installed products updated.

Installed:
ansible-2.9.27-3.el8.noarch      gpm-libs-1.20.7-17.el8.x86_64      libsodium-1.0.18-2.el8.x86_64
python3-bcrypt-3.1.6-2.el8.1.x86_64  python3-jmespath-0.9.0-11.el8.noarch  python3-paramiko-2.4.3-2.el8.noarch
python3-pyasn1-0.3.7-6.el8.noarch    python3-pymacel-1.3.0-5.el8.x86_64    sshpass-1.09-4.el8.x86_64
vim-common-2:8.0.1763-16.el8_5.13.x86_64  vim-enhanced-2:8.0.1763-16.el8_5.13.x86_64  vim-filesystem-2:8.0.1763-16.el8_5.13.noarch
Complete!
```

### Step-4

Download ansible tower archive

**\$ mkdir /tmp/tower**

```
Complete!
[ec2-user@ip-172-31-86-152 ~]$ mkdir /tmp/tower
```

**\$ cd /tmp/tower**

```
[ec2-user@ip-172-31-86-152 ~]$ mkdir /tmp/tower
[ec2-user@ip-172-31-86-152 ~]$ cd /tmp/tower
[ec2-user@ip-172-31-86-152 tower]$
```

We download the latest ansible tower release

\$ curl -k -O <https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-latest.tar.gz>

```
[ec2-user@ip-172-31-86-152 tower]$ curl -k -O https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-latest.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 4621k  100 4621k    0     0  3900k      0  0:00:01  0:00:01 --:--:-- 3900k
```

\$ ls

```
[ec2-user@ip-172-31-86-152 tower]$ ls
ansible-tower-setup-latest.tar.gz
```

Extract download archive

\$ tar xvf ansible-tower-setup-latest.tar.gz

```
ansible-tower-setup-latest.tar.gz
[ec2-user@ip-172-31-86-152 tower]$ tar xvf ansible-tower-setup-latest.tar.gz
ansible-tower-setup-3.8.6-2/
```

Output

```
ansible-tower-setup-3.8.6-2/
ansible-tower-setup-3.8.6-2/collections/
ansible-tower-setup-3.8.6-2/collections/ansible_collections/
ansible-tower-setup-3.8.6-2/collections/ansible_collections/ansible/
ansible-tower-setup-3.8.6-2/collections/ansible_collections/ansible/galaxy_collection/
ansible-tower-setup-3.8.6-2/collections/ansible_collections/ansible/galaxy_collection/roles/
ansible-tower-setup-3.8.6-2/collections/ansible_collections/ansible/galaxy_collection/roles/post_install_config/
ansible-tower-setup-3.8.6-2/collections/ansible_collections/ansible/galaxy_collection/roles/post_install_config/defaults/
```

\$ ls

```
ansible-tower-setup-3.8.6-2/
[ec2-user@ip-172-31-86-152 tower]$ ls
ansible-tower-setup-3.8.6-2  ansible-tower-setup-latest.tar.gz
```

Step-5

Install the ansible tower

Navigate to the created directory

\$ cd ansible-tower-setup-3.8.6-2

```
[ec2-user@ip-172-31-86-152 tower]$
[ec2-user@ip-172-31-86-152 tower]$ cd ansible-tower-setup-3.8.6-2
[ec2-user@ip-172-31-86-152 ansible-tower-setup-3.8.6-2]$
```

Then give the ls command it will shows the directories and files

\$ ls

```
[ec2-user@ip-172-31-86-152 ansible-tower-setup-3.8.6-2]$ ls
backup.yml  group_vars  inventory  README.md  restore.yml  setup.sh
collections  install.yml  licenses  rekey.yml  roles
```

Edit inventory file to set required credentials (Admin password & PG password)

Edit the inventory file

`$ vim inventory`

```
[ec2-user@ip-172-31-86-152 ansible-tower-setup-3.8.6-2]$ vim inventory
```

Inside the inventory file we give the user name and password

```
[tower]
localhost ansible_connection=local

[automationhub]

[database]

[all:vars]
admin_password='admin'
pg_host=''
pg_port=''

pg_database='awx'
pg_username='awx'
pg_password='redhat'
pg_sslmode='prefer' # set to 'verify-full' for client-side enforced SSL

# Automation Hub Configuration
#
automationhub_admin_password=''
automationhub_pg_host=''
automationhub_pg_port=''

automationhub_pg_database='automationhub'
automationhub_pg_username='automationhub'
automationhub_pg_password=''
automationhub_pg_sslmode='prefer'

# By default if the automation hub core and plugin packages
# are installed (i.e. pulp), they won't get upgraded when running the installer
# even if newer packages are available. One needs to run the ./setup.sh
# script with the following set to True.
#
automationhub_upgrade = False

# By default, the Ansible package will not be upgraded
# to the latest version, even if one exists in the bundled
# installer or another repository. Set upgrade_ansible_with_hub
# to True if you want Ansible to be upgraded
```

Step-6

Installation of ansible tower

`$ sudo ./setup.sh`

```
The setup process completed successfully.
Setup log saved to /var/log/tower/setup-2022-05-19-09:09:15.log.
[ec2-user@ip-172-31-86-152 ansible-tower-setup-3.8.6-2]$
```

`$ sudo ansible-tower-service start`

```
[ec2-user@ip-172-31-86-152 ansible-tower-setup-3.8.6-2]$ sudo ansible-tower-service start
```

`$ sudo ansible-tower-service status`

```
[ec2-user@ip-172-31-86-152 ansible-tower-setup-3.8.6-2]$ sudo ansible-tower-service status
• ansible-tower.service - Ansible Tower service
  Loaded: loaded (/usr/lib/systemd/system/ansible-tower.service; enabled; vendor preset: disabled)
  Active: active (exited) since Thu 2022-05-19 09:17:53 UTC; 2min 53s ago
  Process: 77406 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
  Main PID: 77406 (code=exited, status=0/SUCCESS)

May 19 09:17:53 ip-172-31-86-152.ec2.internal systemd[1]: Starting Ansible Tower service...
May 19 09:17:53 ip-172-31-86-152.ec2.internal systemd[1]: Started Ansible Tower service.

• postgresql.service - PostgreSQL database server
  Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/postgresql.service.d
           └─override.conf
  Active: active (running) since Thu 2022-05-19 09:10:17 UTC; 10min ago
  Main PID: 71510 (postmaster)
  Tasks: 17 (limit: 49245)
```

After run the script it will shows

The setup process completed successfully

After open the browser we give the ip address

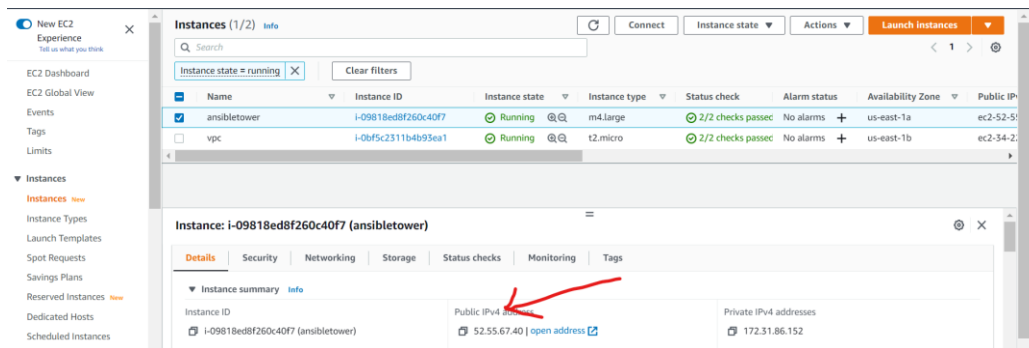
It will ask the User name and password

Username: admin

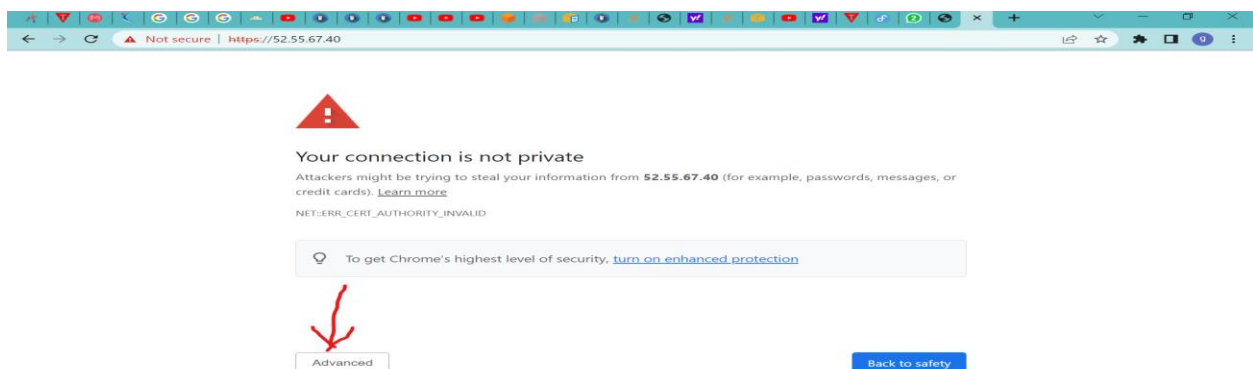
Password: redhat

Step-7

Lets check that by taking the public ip of the instance



Paste the ip address in the web browser the output will come like this ( we need to select the advance option )

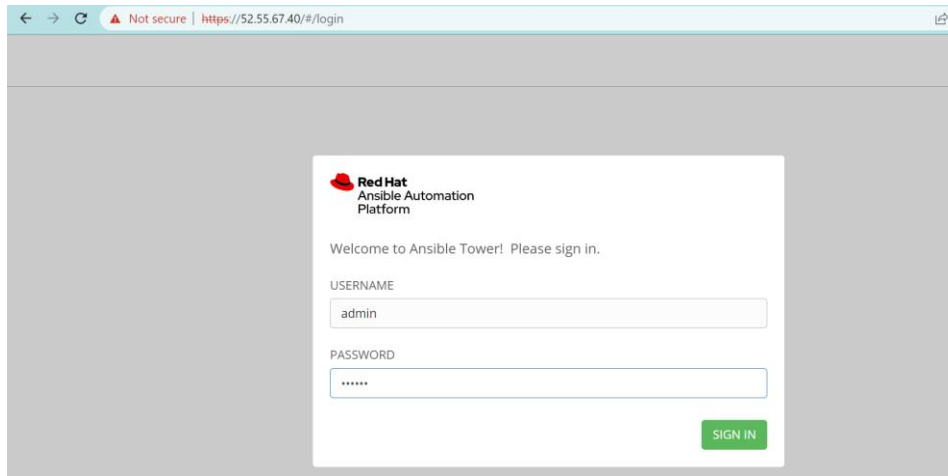


We need to proceed that

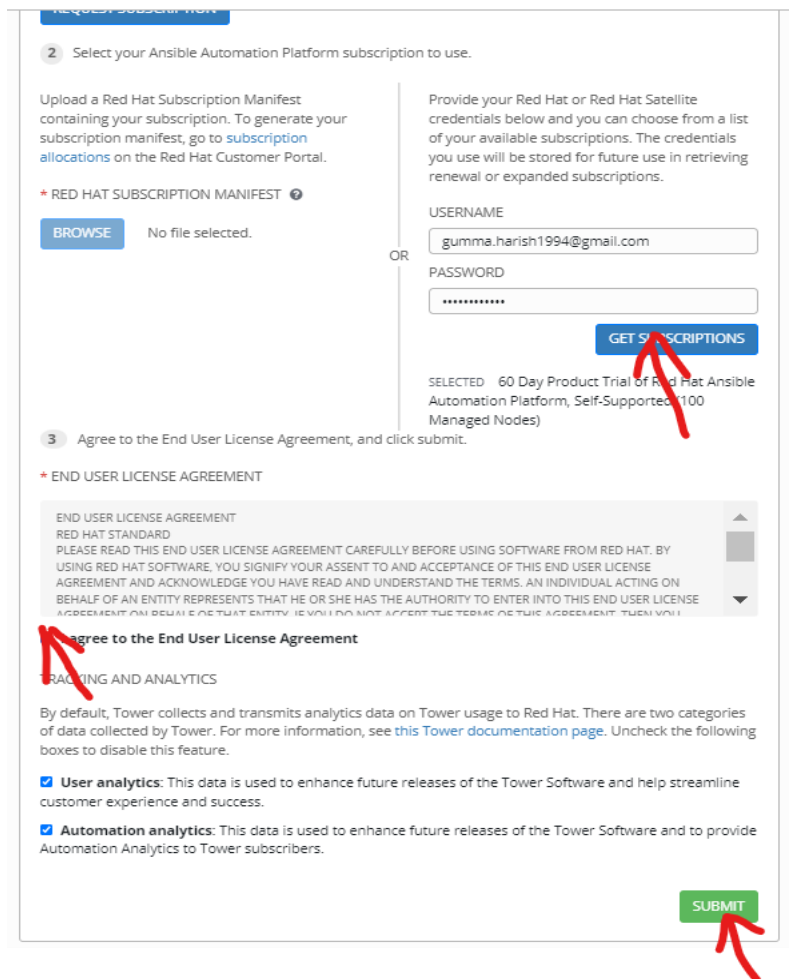




Now we got the login page



After login the page we get the this page



Finally we will get ansible tower dashboard

