

Animated It Infrastructure Monitoring with Ansible Prometheus and Grafana

Step 1: Login to AWS.com

sign-in.aws.amazon.com/sign-in?redirect_uri=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fconsole%2Fhome%3FhashArgs%3D%2523%26isauthcode%3Dtrue%26region%3Dus-east-1%26state%3Dhas...

Try the new sign in UI

See our new improved Amazon Web Services sign in experience before we officially launch.

Enable new sign in

aws

Sign in

☒ Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ IAM user

User within an account that performs daily tasks. [Learn more](#)

Root user email address

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

☐ New to AWS?

Create a new AWS account

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Multi-session disabled

English

Transform financial services

Improve business and operational agility with cloud solutions from AWS

Download the ebook

Step 2: Once logged in click on the grid of dots in the top left and select all services and go down to EC2 and click on it

The screenshot displays the AWS Management Console interface. On the left, a dark navigation sidebar lists various AWS services. The 'EC2' service is highlighted with a blue star icon and the text 'Virtual Servers in the Cloud'. The main content area on the right shows the 'Applications (0)' dashboard for the 'us-east-1 (Current Region)' region. It includes a search bar, a table with columns for Name, Description, Region, and Origin, and a 'Create application' button. Below this, there's a 'Cost and usage' section with a bar chart showing current month costs (\$0.00) and a 'Security' section for the 'us-east-1 (N. Virginia)' region. The bottom of the page features a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc.

Step 3: Once we're on the EC2 page we can click on launch instance to create our EC2's

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Overview

EC2

Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

▼ Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

Load Balancers

Target Groups

Trust Stores

▼ Auto Scaling

Auto Scaling Groups

Settings

Compute

Amazon Elastic Compute Cloud (EC2)

Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

Launch a virtual server

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#)

[View dashboard](#)

Benefits and features

EC2 offers ultimate scalability and control

Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Widest variety of server size options
- Widest availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

[Find out more about EC2](#)

Use cases

Run cloud-native and enterprise applications

Amazon EC2 delivers secure, reliable, high-performance, and cost-effective compute infrastructure to meet demanding business needs.

Scale for HPC applications

Access the on-demand infrastructure and capacity you need to run HPC applications faster and cost-effectively.

Train and deploy ML applications

Amazon EC2 delivers the broadest choice of compute, networking (up to 400 Gbps), and storage services purpose-built to optimize price performance for ML projects.

Related services

[Lightsail](#)

[Lambda](#)

Get started

Take our walkthroughs to help you launch an instance, learn about EC2 best practices, and set up your account.

[Get started walkthroughs](#)

[Get started tutorial](#)

Additional actions

[View running instances](#)

[Migrate a server](#)

[Create load balancer](#)

[Request Spot Instances](#)

[Create an Auto Scaling group](#)

[Browse images \(AMIs\)](#)

[Manage volumes](#)

[Manage snapshots](#)

[Manage firewalls \(security groups\)](#)

Pricing (US)

[EC2 pricing options](#)

Step 4: Create EC2 Instances on AWS we will need 4 and we will use Ubuntu

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices. Do not show me this message again Take a walkthrough

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: Add additional tags

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-0e1bed4f06a3b463d (64-bit (x86)) / ami-0485c8c0dadf51233 (64-bit (ARM))
Virtualization: hvm ENA enabled: true Root device type: ebs Free tier eligible

Description

Ubuntu Server 22.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Canonical, Ubuntu, 22.04, amd64 jammy image

Architecture	AMI ID	Username
64-bit (x86)	ami-0e1bed4f06a3b463d	ubuntu Verified provider

Instance type

Instance type: Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour

All generations Compare instance types

Summary

Number of instances: Info

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)
Canonical, Ubuntu, 22.04, amd64...read more
ami-0e1bed4f06a3b463d

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance Preview code

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 5: Create ppk and security groups

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

EC2 > Instances > Launch an instance

▼ Instance type

Instance type

t2.micro Free tier eligible All generations [Compare instance types](#)

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

[Additional costs apply for AMIs with pre-installed software](#)

▼ Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

revatureppk [Create new key pair](#)

▼ Network settings

Network vpc-0772bdf27c936d233

Subnet No preference (Default subnet in any availability zone)

Auto-assign public IP Enable [Additional charges apply when outside of free tier allowance](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Create security group ☒ Select existing security group

Common security groups [Info](#)

Select security groups

ansible-new-sg sg-0b9f47e4424691083 [Compare security group rules](#)

VPC: vpc-0772bdf27c936d233

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Configure storage

1x 8 GiB gp2 [Advanced](#)

Root volume, Not encrypted

[Free tier eligible customers can get up to 30 GiB of EBS General Purpose \(SSD\) or Magnetic storage](#)

[Add new volume](#)

▼ Summary

Number of instances [Info](#)

4

When launching more than 1 instance, consider [EC2 Auto Scaling](#)

Software Image (AMI)

Canonical, Ubuntu, 22.04, amd64...[read more](#)
ami-0e1bed4f06a35b463d

Virtual server type (instance type)

t2.micro

Firewall (security group)

ansible-new-sg

Storage (volumes)

1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Step 6: Use putty to get to the command line of the EC2's

Instances (1/4) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
machine1	i-0d9b9a6ee85c59800	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-3-82-176-179.com...	3.82.176.17
controller	i-0c3230085e9f1cced	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-54-234-40-20.com...	54.234.40.2
machine2	i-04d3f3a419c8f467e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-100-27-19-86.com...	100.27.19.8
machine3	i-0b05b66ecb8c4d578	Initializing	t2.micro	Initializing	View alarms +	us-east-1c	ec2-44-202-18-193.co...	44.202.18.1

PuTTY Configuration

Category: Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) 44.202.18.193 Port 22

Connection type: SSH Serial Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings Load Save Delete

Close window on exit: Always Never Only on clean exit

About Help Open Cancel

Instance summary Info

Instance ID i-0b05b66ecb8c4d578

IPv6 address -

Hostname type IP name: ip-172-31-93-20.ec2.internal

Public IPv4 address copied

Public IPv4 address 44.202.18.193 | open address

Instance state Running

Private IP DNS name (IPv4 only) ip-172-31-93-20.ec2.internal

Private IPv4 addresses 172.31.93.20

Public IPv4 DNS ec2-44-202-18-193.compute-1.amazonaws.com | open address

PuTTY Configuration

Category: Credentials to authenticate with

Public-key authentication

Private key file for authentication: Browse...

Certificate to use with the private key (optional): Browse...

Plugin to provide authentication responses

Plugin command to run

Select private key file

File Explorer

2/28/2025 10:55 AM PuTTY Private Key ... 2 KB

Step 7: Create names for each instance. `sudo hostname (name)`

```
ubuntu@ip-172-31-93-20: ~  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-93-20:~$ sudo hostname machine3
```

Step 8: sudo su to apply name

```
root@machine3: /home/ubuntu  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-93-20:~$ sudo hostname machine3  
ubuntu@ip-172-31-93-20:~$ sudo su  
root@machine3:/home/ubuntu#
```

Step 9: Add the private ip addresses of all EC2 in the network. nano /etc/hosts

```
GNU nano 6.2 /etc/hosts *
172.31.86.134 controller
172.31.85.124 machine1
172.31.88.65 machine2
172.31.93.20 machine3
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

Step 10: Change login permissions. nano /etc/ssh/sshd_config PermitRootLogin yes
PasswordAuthentication yes #KbdInteractiveAuthentication no


```
root@machine3: /etc/ssh
GNU nano 6.2 sshd config *
# This ssnd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
#KbdInteractiveAuthentication no

# Kerberos options
^G Help      ^O Write Out  ^W Where Is   ^E Cut        ^T Execute    ^C Location   M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-R Redo     M-C Copy
```

Step 11: Restart sshd service. service sshd restart

```
root@machine3: /etc/ssh
grub.d                nsswitch.conf        update-manager
gshadow               opt                  update-motd.d
gshadow-              os-release           update-notifier
gss                   overlayroot.conf     usb_modeswitch.conf
hdparm.conf           overlayroot.local.conf  usb_modeswitch.d
hibagent-config.cfg   pam.conf             vim
hibinit-config.cfg    pam.d               vmware-tools
host.conf             passwd              vtrgb
hostname              passwd-             wgetrc
hosts                 perl                xattr.conf
hosts.allow           pm                   xdg
hosts.deny            polkit-1            zsh_command_not_found
init.d               pollinate
initramfs-tools       ppp
root@machine3:/etc# cd /.ssh
bash: cd: /.ssh: No such file or directory
root@machine3:/etc# cd /ssh
bash: cd: /ssh: No such file or directory
root@machine3:/etc# nano sshd_config
root@machine3:/etc# cd ssh
root@machine3:/etc/ssh# ls
moduli      ssh_host_ecdsa_key    ssh_host_ed25519_key.pub  ssh_import_id
ssh_config  ssh_host_ecdsa_key.pub ssh_host_rsa_key           sshd_config
ssh_config.d ssh_host_ed25519_key  ssh_host_rsa_key.pub      sshd_config.d
root@machine3:/etc/ssh# nano sshd_config
root@machine3:/etc/ssh# service sshd restart
root@machine3:/etc/ssh# passwd
New password:
Retype new password:
passwd: password updated successfully
root@machine3:/etc/ssh#
```

Step 12: Set password for machines passwd

Step 13: Now we can ssh from one machine into another

Step 14: Move to .ssh cd /root/.ssh

Step 15: Create public/private rsa key pair. ssh-keygen

```
root@machine3: ~/.ssh
root@machine3:/etc/ssh# nano sshd_config
root@machine3:/etc/ssh# service sshd restart
root@machine3:/etc/ssh# passwd
New password:
Retype new password:
passwd: password updated successfully
root@machine3:/etc/ssh# cd /root/.ssh
root@machine3:~/.ssh# ls
authorized_keys
root@machine3:~/.ssh# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1Nadh0aa8Clf1NNCNbaQ5k7/FM1BXA/MbPkWn126gFQ root@machine3
The key's randomart image is:
+----[RSA 3072]-----+
|      .  . =OBoo|
|      =EB* =BB.|
|      =.*o*++oB|
|      o.o oo..+*|
|      .S..o o o.|
|      . . o . .|
|      . . o . .|
|      . . .|
|      . . .|
+-----[SHA256]-----+
root@machine3:~/.ssh#
```

Step 16: Copy keys to other machines. ssh-copy-id root@machine1

```
root@machine3: ~/.ssh
-----[RSA 3072]-----
.  =OBoo|
.  =EB*=BB.|
.  =.*o*++oB|
.  o.o oo..+*|
.  S..o o o.|
.  . o . .|
.  . o |
.  . |
-----[SHA256]-----
root@machine3:~/.ssh# ssh-copy-id root@machine1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'machine1 (172.31.85.124)' can't be established.
ED25519 key fingerprint is SHA256:VDnWODN0YbEEF8/Ly2cptk8k8kQgocmnR/Niy4vm6BpI.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
(root@machine1) Password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@machine1'"
and check to make sure that only the key(s) you wanted were added.

root@machine3:~/.ssh# ssh root@machine1
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Mar  2 18:04:38 UTC 2025

System load:  0.0               Processes:           111
Usage of /:   38.2% of 7.57GB   Users logged in:    1
Memory usage: 31%              IPv4 address for eth0: 172.31.85.124
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

21 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Sun Mar  2 17:55:45 2025 from 172.31.86.134
root@machine1:~# exit
logout
Connection to machine1 closed.
root@machine3:~/.ssh#
```

Step 17: Now we can ssh into the other machines without entering a password

Step 18: Install Ansible in the controller

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

```
root@controller: /home/ubuntu
root@controller:/home/ubuntu# sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Step 19: Check version of Ansible. `ansible -- version`

Step 20: Create file to point to EC2s nano `ansiblevms`

```
GNU nano 6.2 ansiblevms
[all]
controller
machine1
machine2
machine3

[con]
controller

[mach]
machine1
machine2
machine3

[ Read 13 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste       ^J Justify    ^_ Go To Line  M-E Redo
```

Step 21: Make vm file an inventory file and use ping module to ping our EC2
`ansible -i ansiblevms -m ping mach`

Step 22: Disable host key checking in controller `cd /etc/ssh nano ssh_config`
`StrictHostKeyChecking no`

```
root@controller:/home/ubuntu
path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for
more information.
controller | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.10"
  },
  "changed": false,
  "ping": "pong"
}
root@controller:/home/ubuntu# nano /etc/ssh/ssh_config
GNU nano 6.2 /etc/ssh/ssh_config

# This is the ssh client system-wide configuration file. See
# ssh_config(5) for more information. This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Include /etc/ssh/ssh_config.d/*.conf

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# BatchMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
StrictHostKeyChecking no
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519
# Port 22
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
```

Step 23: Disable HostBasedAuthentication `cd /etc/ssh nano sshd_config`
`HostBasedAuthentication no`

```
root@controller: /home/ubuntu
root@controller:/home/ubuntu# nano /etc/ssh/ssh_config
root@controller:/home/ubuntu# nano /etc/ssh/sshd_config
GNU nano 6.2 /etc/ssh/sshd_config *

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
```

Step 24: APT update for controller. apt update

```
root@controller: /home/ubuntu
/usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that
path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for
more information.
controller | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.10"
  },
  "changed": false,
  "ping": "pong"
}
root@controller:/home/ubuntu# nano /etc/ssh/ssh_config
root@controller:/home/ubuntu# nano /etc/ssh/sshd_config
root@controller:/home/ubuntu# apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Fetched 129 kB in 1s (169 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
21 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@controller:/home/ubuntu# apt install -y mysql-server python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot g++ g++-11 gcc gcc-11 gcc-11-base javascript-common
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1
```

Step 25: Install mysql server. `apt install -y mysql-server python3-pip`

Step 26: Check status of my sql `systemctl status mysql`

```
root@controller: /home/ubuntu
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@controller:/home/ubuntu# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-03-02 18:35:41 UTC; 2min 18s ago
     Main PID: 32610 (mysqld)
    Status: "Server is operational"
       Tasks: 37 (limit: 1130)
      Memory: 349.8M
         CPU: 1.383s
    CGroup: /system.slice/mysql.service
           └─32610 /usr/sbin/mysqld

Mar 02 18:35:40 controller systemd[1]: Starting MySQL Community Server...
Mar 02 18:35:41 controller systemd[1]: Started MySQL Community Server.
root@controller:/home/ubuntu# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Step 27: Create username and password for logging into the server. `mysql -u root -p`

Step 28: Create Database and users for whatever machine we might use and also give them privileges.

```
root@controller: /home/ubuntu
CGroup: /system.slice/mysql.service
        └─32610 /usr/sbin/mysqld

Mar 02 18:35:40 controller systemd[1]: Starting MySQL Community Server...
Mar 02 18:35:41 controller systemd[1]: Started MySQL Community Server.
root@controller:/home/ubuntu# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.41-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database system_metrics
-> ;
Query OK, 1 row affected (0.04 sec)

mysql> create user 'metrics_user'@'%' identified by '123';
Query OK, 0 rows affected (0.03 sec)

mysql> grant all privileges on system_metrics.* to 'metrics_user'@'%' with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Step 29: Flush privileges. `flush privileges;`

Step 30: Change bind address in `/etc/mysql/mysql.conf.d/ mysqld.cnf`

```
root@controller: /etc/mysql/mysql.conf.d
mysql> exit
Bye
root@controller:/home/ubuntu# cd /etc/mysql/mysql.conf.d/
root@controller:/etc/mysql/mysql.conf.d# ls
mysql.cnf  mysqld.cnf
root@controller:/etc/mysql/mysql.conf.d# nano mysqld.cnf
GNU nano 6.2 mysqld.cnf *
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket              = /var/run/mysqld/mysqld.sock
# port                = 3306
# datadir             = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir              = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 0.0.0.0
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
```

Step 31: Restart mysql. systemctl restart mysql

Step 32: Create ansible playbook. nano metrics.yaml


```
root@controller: /home/ubuntu
GNU nano 6.2 metrics.yaml
--
- name: Setup Linux Metrics Collection on Multiple Servers
  hosts: mach
  become: yes
  tasks:
    - name: Install required packages
      apt:
        name:
          - python3
          - python3-pip
        state: present

    - name: Install Python dependencies
      pip:
        name:
          - psutil
          - mysql-connector-python

    - name: Deploy Python Script for Metrics Collection
      copy:
        dest: /opt/linux_metrics.py
        mode: "0755"
        content: |
          import psutil
          import mysql.connector
          from datetime import datetime
          import socket

          # Database Configuration (Central MySQL Server)
          db_config = {
            "host": "172.31.86.134", # Change to Ansible Controller's IP
            "user": "metrics_user",
            "password": "123",
            "database": "system_metrics"
          }

          def get_system_metrics():
            cpu_usage = psutil.cpu_percent(interval=1)
            memory_usage = psutil.virtual_memory().percent
            disk_usage = psutil.disk_usage('/').percent
            hostname = socket.gethostname()
            return hostname, cpu_usage, memory_usage, disk_usage

          def insert_into_db(hostname, cpu, memory, disk):
            try:
              conn = mysql.connector.connect(**db_config)
              cursor = conn.cursor()
              sql = "INSERT INTO metrics (server_name, cpu_usage, memory_usage, disk_usage) VALUES (%s, %s, %s, %s)"
              cursor.execute(sql, (hostname, cpu, memory, disk))
              conn.commit()
              cursor.close()
            except:
              conn.close()
```

Step 33: Check the playbook and see if we need to install packages. `Ansible-playbook -i hosts metrics.yaml -- check`

```
root@controller: /home/ubuntu
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
root@controller: /home/ubuntu# ansible-playbook -i hosts metrics.yaml

PLAY [Setup Linux Metrics Collection on Multiple Servers] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host machine1 is using the discovered Python interpreter at
/usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [machine1]
[WARNING]: Platform linux on host machine2 is using the discovered Python interpreter at
/usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [machine2]
[WARNING]: Platform linux on host machine3 is using the discovered Python interpreter at
/usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [machine3]

TASK [Install required packages] *****
changed: [machine2]
changed: [machine1]
changed: [machine3]

TASK [Install Python dependencies] *****
changed: [machine1]
changed: [machine2]
changed: [machine3]

TASK [Deploy Python Script for Metrics Collection] *****
changed: [machine3]
changed: [machine1]
changed: [machine2]

TASK [Setup Cron Job for Data Collection] *****
changed: [machine3]
changed: [machine2]
changed: [machine1]

PLAY RECAP *****
machine1      : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
machine2      : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
machine3      : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
root@controller: /home/ubuntu#
```

Step 34: Apt update the machines to make sure we have the packages needed. `ansible -i hosts mach -a 'apt update'`

Step 35: Log into mysql and show databases

```
root@controller:/home/ubuntu# mysql -u metrics_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.41-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
| system_metrics |
+-----+
3 rows in set (0.02 sec)

mysql>
```

Step 36: Create Table metrics and SELECT * FROM metrics

```
mysql> use system_metrics
Database changed
mysql> create table if not exists metrics( id INT AUTO_INCREMENT PRIMARY KEY, timestamp DATETIME DEFAULT CURRENT_TIMESTAMP, server_name varchar(255), cpu_usage float, memory_usage float, disk_usage float );
Query OK, 0 rows affected (0.06 sec)

mysql>
```

```
mysql> SELECT * FROM metrics ORDER BY timestamp DESC;
+-----+-----+-----+-----+-----+-----+
| id | timestamp | server_name | cpu_usage | memory_usage | disk_usage |
+-----+-----+-----+-----+-----+-----+
| 7 | 2025-03-02 19:24:02 | machine3 | 18 | 37 | 37.2 |
| 8 | 2025-03-02 19:24:02 | machine2 | 5.7 | 40.4 | 43.9 |
| 9 | 2025-03-02 19:24:02 | machine1 | 0 | 38.8 | 43.8 |
| 5 | 2025-03-02 19:23:03 | machine3 | 12.2 | 37 | 37.2 |
| 6 | 2025-03-02 19:23:03 | machine2 | 16.7 | 40.4 | 43.9 |
| 4 | 2025-03-02 19:23:02 | machine1 | 1 | 38.8 | 43.8 |
| 1 | 2025-03-02 19:22:02 | machine1 | 0 | 38.9 | 43.8 |
| 2 | 2025-03-02 19:22:02 | machine3 | 29.1 | 37 | 37.2 |
| 3 | 2025-03-02 19:22:02 | machine2 | 5.7 | 40.4 | 43.9 |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)

mysql>
```

Step 37: Download prometheus using wget and the download link.

```
root@controller:/home/ubuntu# wget https://github.com/prometheus/prometheus/releases/download/v2.53.3/prometheus-2.53.3.linux-amd64.tar.gz
```

Step 38: unzip and move the downloaded files into a prometheus directory

```

root@controller:/home/ubuntu# tar xvfz prometheus-2.53.3.linux-amd64.tar.gz
prometheus-2.53.3.linux-amd64/
prometheus-2.53.3.linux-amd64/consoles/
prometheus-2.53.3.linux-amd64/consoles/node-disk.html
prometheus-2.53.3.linux-amd64/consoles/node-overview.html
prometheus-2.53.3.linux-amd64/consoles/prometheus.html
prometheus-2.53.3.linux-amd64/consoles/index.html.example
prometheus-2.53.3.linux-amd64/consoles/node-cpu.html
prometheus-2.53.3.linux-amd64/consoles/prometheus-overview.html
prometheus-2.53.3.linux-amd64/consoles/node.html
prometheus-2.53.3.linux-amd64/promtool
prometheus-2.53.3.linux-amd64/LICENSE
prometheus-2.53.3.linux-amd64/prometheus
prometheus-2.53.3.linux-amd64/prometheus.yml
prometheus-2.53.3.linux-amd64/console_libraries/
prometheus-2.53.3.linux-amd64/console_libraries/menu.lib
prometheus-2.53.3.linux-amd64/console_libraries/prom.lib
prometheus-2.53.3.linux-amd64/NOTICE
root@controller:/home/ubuntu# ls
ansible      hosts      prometheus-2.53.3.linux-amd64  user.yaml
ansiblevms  metrics.yaml  prometheus-2.53.3.linux-amd64.tar.gz
root@controller:/home/ubuntu# rm prometheus-2.53.3.linux-amd64.tar.gz
root@controller:/home/ubuntu# mv prometheus-2.53.3.linux-amd64/ prometheus
root@controller:/home/ubuntu# ls
ansible  ansiblevms  hosts  metrics.yaml  prometheus  user.yaml
root@controller:/home/ubuntu# clear

```

Step 39: Steps on prometheus website. `sudo useradd --no-create-home --shell`

`/bin/false prometheus`

`sudo mkdir /etc/prometheus`

`sudo mkdir /var/lib/prometheus`

`sudo chown prometheus:prometheus /etc/prometheus`

`sudo chown prometheus:prometheus /var/lib/prometheus`

`sudo cp prometheus/prometheus /usr/local/bin/`

`sudo cp prometheus/promtool /usr/local/bin/`

`sudo chown prometheus:prometheus /usr/local/bin/prometheus`

`sudo chown prometheus:prometheus /usr/local/bin/promtool`

`sudo cp -r prometheus/consoles /etc/prometheus`

`sudo cp -r prometheus/console_libraries /etc/prometheus`

```
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
```

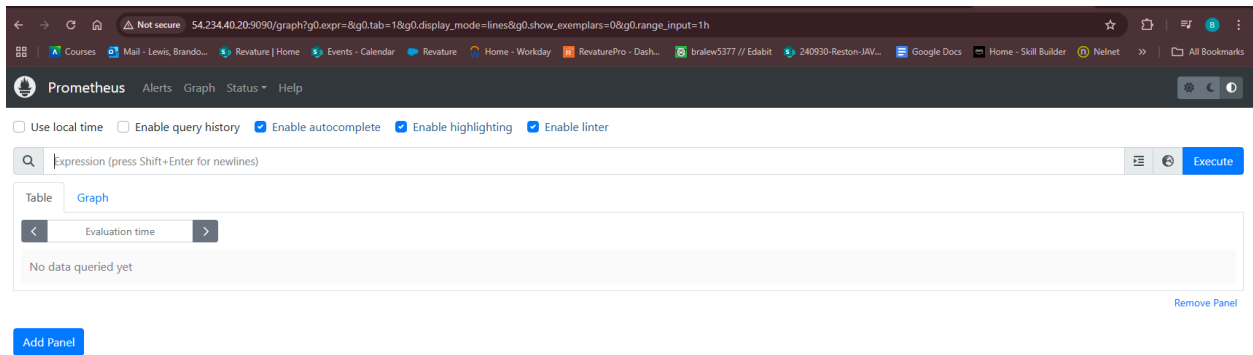
```
sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
```

Step 40: Create service file `sudo nano /etc/systemd/system/prometheus.service`

Step 41: Copy files from `home/ubuntu/prometheus` to `/etc/prometheus` then Daemon reload and start prometheus then check status

```
root@controller:/home/ubuntu# cd prometheus
root@controller:/home/ubuntu/prometheus# ls
LICENSE NOTICE console_libraries consoles prometheus prometheus.yml promtool
root@controller:/home/ubuntu/prometheus# cp -r * /etc/prometheus
root@controller:/home/ubuntu/prometheus# systemctl daemon-reload
root@controller:/home/ubuntu/prometheus# systemctl start prometheus
root@controller:/home/ubuntu/prometheus# systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; disabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-03-02 19:58:23 UTC; 12s ago
 Main PID: 33461 (prometheus)
    Tasks: 6 (limit: 1130)
   Memory: 31.2M
      CPU: 66ms
   CGroup: /system.slice/prometheus.service
           └─33461 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb.path /va>
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.114Z caller=tls_config.go:316 level=info
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.114Z caller=head.go:793 level=info compo
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.114Z caller=head.go:830 level=info compo
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.120Z caller=main.go:1169 level=info fs t
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.120Z caller=main.go:1172 level=info msg=>
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.120Z caller=main.go:1354 level=info msg=>
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.123Z caller=main.go:1391 level=info msg=>
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.124Z caller=main.go:1402 level=info msg=>
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.124Z caller=main.go:1133 level=info msg=>
Mar 02 19:58:24 ip-172-31-86-134 prometheus[33461]: ts=2025-03-02T19:58:24.125Z caller=manager.go:164 level=info co
root@controller:/home/ubuntu/prometheus#
```

Step 42: With prometheus running go to browser and enter ip address of controller with :9090 at the end



Step 43: Create ansible playbook to install node exporter on all of the machines and run it.

```
root@controller:/home/ubuntu# nano node.yaml
root@controller:/home/ubuntu# ansible-playbook -i hosts node.yaml --check

PLAY [Install and Configure Prometheus Node Exporter on Ubuntu] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host machine3 is using the discovered Python interpreter at /usr/bin/python3.10, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [machine3]
[WARNING]: Platform linux on host machine2 is using the discovered Python interpreter at /usr/bin/python3.10, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [machine2]
[WARNING]: Platform linux on host machine1 is using the discovered Python interpreter at /usr/bin/python3.10, but
future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [machine1]

TASK [Update package lists] *****
changed: [machine2]
changed: [machine1]
changed: [machine3]

TASK [Install required dependencies] *****
ok: [machine2]
ok: [machine3]
ok: [machine1]

TASK [Download Node Exporter] *****
changed: [machine1]
changed: [machine2]
changed: [machine3]

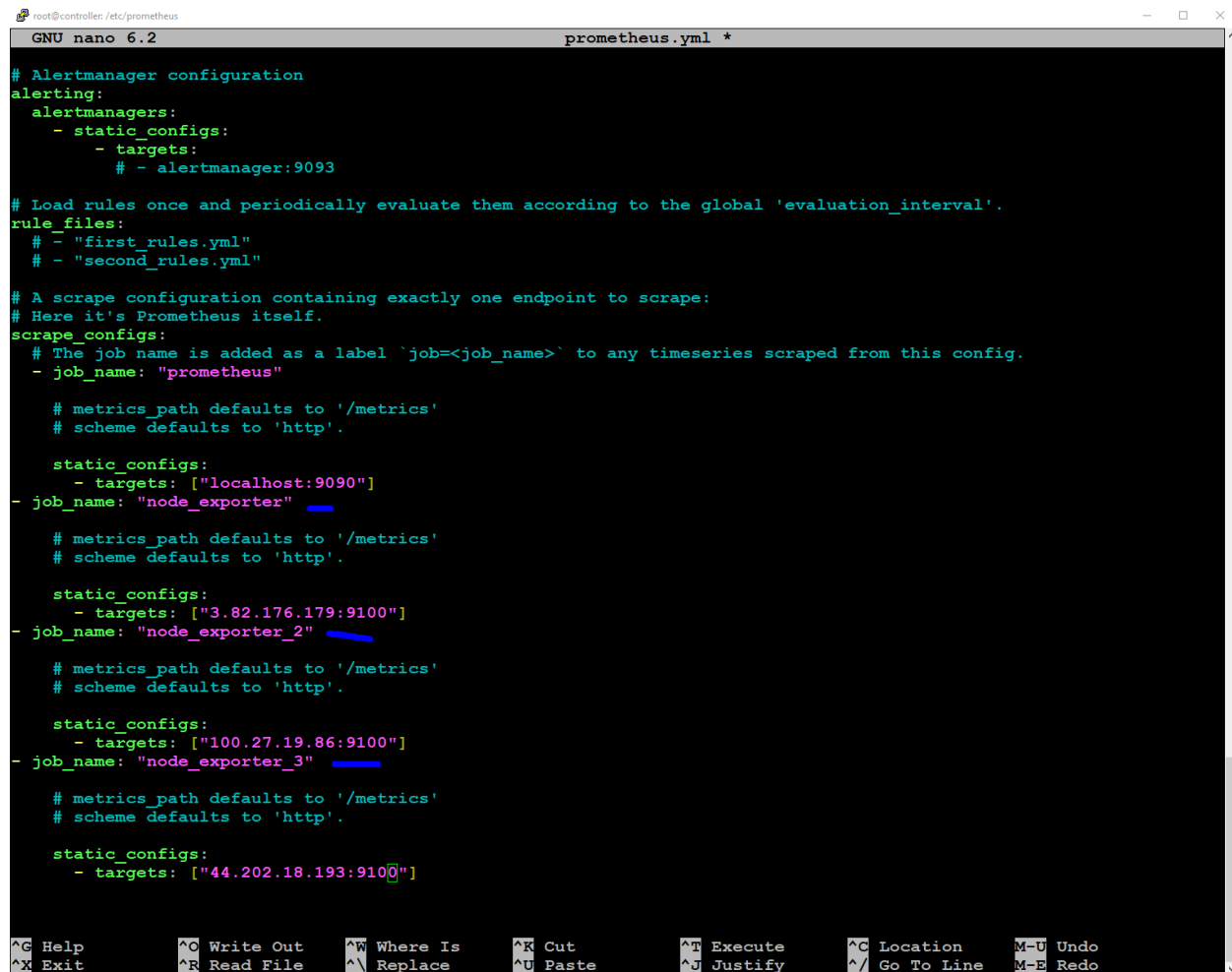
TASK [Extract Node Exporter] *****
fatal: [machine2]: FAILED! => {"changed": false, "msg": "Source '/tmp/node_exporter.tar.gz' does not exist"}
fatal: [machine1]: FAILED! => {"changed": false, "msg": "Source '/tmp/node_exporter.tar.gz' does not exist"}
fatal: [machine3]: FAILED! => {"changed": false, "msg": "Source '/tmp/node_exporter.tar.gz' does not exist"}

PLAY RECAP *****
machine1      : ok=4    changed=2    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
machine2      : ok=4    changed=2    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
machine3      : ok=4    changed=2    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

root@controller:/home/ubuntu# ansible-playbook -i hosts node.yaml

PLAY [Install and Configure Prometheus Node Exporter on Ubuntu] *****
```

Step 44: In /etc/prometheus/prometheus.yml change job name and local host to controller and machines.



```
GNU nano 6.2 prometheus.yml *
# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
- job_name: "node_exporter"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["3.82.176.179:9100"]
- job_name: "node_exporter_2"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["100.27.19.86:9100"]
- job_name: "node_exporter_3"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["44.202.18.193:9100"]

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

Step 45: Restart prometheus. systemctl restart prometheus

Step 46: Download prometheus alert manager using wget and the download link

```
root@controller: /home/ubuntu
root@controller:/home/ubuntu# wget https://github.com/prometheus/alertmanager/releases/download/v0.28.0/alertmanager-0.28.0.linux-amd64.tar.gz
```

Step 47: Run these commands

```
sudo groupadd -f alertmanager
sudo useradd -g alertmanager --no-create-home --shell /bin/false
alertmanager
sudo mkdir -p /etc/alertmanager/templates
sudo mkdir /var/lib/alertmanager
sudo chown alertmanager:alertmanager /etc/alertmanager
sudo chown alertmanager:alertmanager /var/lib/alertmanager
sudo cp alertmanager/alertmanager /usr/bin/
sudo cp alertmanager/amtool /usr/bin/
sudo chown alertmanager:alertmanager /usr/bin/alertmanager
sudo chown alertmanager:alertmanager /usr/bin/amtool

sudo cp alertmanager/alertmanager.yml
/etc/alertmanager/alertmanager.yml

sudo chown alertmanager:alertmanager
/etc/alertmanager/alertmanager.yml

sudo nano /usr/lib/systemd/system/alertmanager.service
```

Step 48: copy contents into alertmanager.service


```
root@controller: /home/ubuntu
GNU nano 6.2 /usr/lib/systemd/system/alertmanager.service

[Unit]
Description=AlertManager
Wants=network-online.target
After=network-online.target

[Service]
User=alertmanager
Group=alertmanager
Type=simple
ExecStart=/usr/bin/alertmanager \
    --config.file /etc/alertmanager/alertmanager.yml \
    --storage.path /var/lib/alertmanager/

[Install]
WantedBy=multi-user.target

[ Read 15 lines ]
^G Help      ^C Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^/_ Go To Line M-E Redo
```

Step 49: Reload Daemon and start alertmanager

Step 50: In /etc/prometheus make a directory called rules and add the rules.yaml file

```
root@controller: /etc/prometheus
Mar 03 06:11:57 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:57.623Z caller=main.go:216 msg=>
Mar 03 06:11:57 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:57.623Z caller=main.go:217 build=>
Mar 03 06:11:57 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:57.624Z caller=cluster.go:161 c=>
Mar 03 06:11:57 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:57.664Z caller=cluster.go:623 c=>
Mar 03 06:11:57 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:57.678Z caller=coordinator.go:1=>
Mar 03 06:11:57 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:57.679Z caller=coordinator.go:1=>
Mar 03 06:11:57 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:57.685Z caller=main.go:485 msg=>
Mar 03 06:11:59 ip-172-31-86-134 alertmanager[35258]: level=info ts=2025-03-03T06:11:59.664Z caller=cluster.go:648 c=>

root@controller:/home/ubuntu# cd /etc/prometheus
root@controller:/etc/prometheus# ls
LICENSE NOTICE  console libraries  consoles  prometheus  prometheus.yml  promtool
root@controller:/etc/prometheus# mkdir rules
root@controller:/etc/prometheus# cd rules/
root@controller:/etc/prometheus/rules# ls
root@controller:/etc/prometheus/rules# nano rules.yaml
root@controller:/etc/prometheus/rules# cd ..
root@controller:/etc/prometheus# nano prometheus.yml
root@controller:/etc/prometheus# cd rules
root@controller:/etc/prometheus/rules# ls
rules.yaml
root@controller:/etc/prometheus/rules# nano rules
root@controller:/etc/prometheus/rules# nano rules.yaml
root@controller:/etc/prometheus/rules# cd ..
root@controller:/etc/prometheus# ./promtool check rules rules/rules.yaml
Checking rules/rules.yaml
SUCCESS: 3 rules found
```

Step 51: Use promtool to see how many rules we have ./promtool check rules rules/rules.yaml

Step 52: In prometheus.yml remove the # from first rules and name the location of our rule file

```
root@controller: /etc/prometheus
GNU nano 6.2 prometheus.yml *
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            - 54.234.40.20:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  - "rules/rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'

^G Help      ^O Write Out  ^W Where Is   ^R Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^/ Go To Line  M-E Redo
```

Step 53: restart prometheus and alertmanager and go to controller ip address plus :9093 in browser

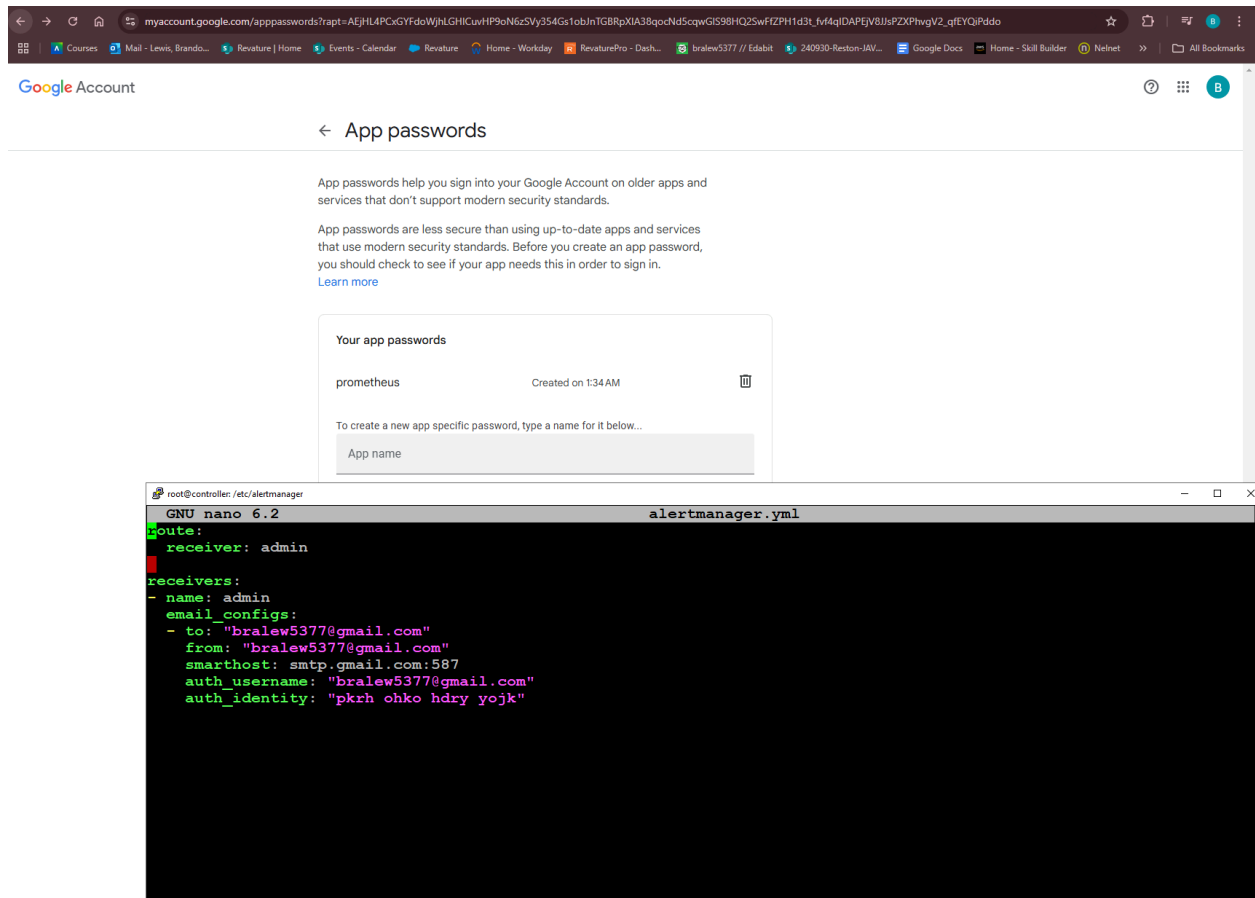
Step 54: add rule in rules file

```
root@controller: /etc/prometheus/rules
GNU nano 6.2 rules.yml
groups:
  - name: my-rule
    rules:
      - alert: Node_exporter_down
        expr: up{job="node_exporter"} == 0
        for: 1m
      - alert: Node_exporter_2_down
        expr: up{job="node_exporter_2"} == 0
        for: 1m
      - alert: Node_exporter_3_down
        expr: up{job="node_exporter_3"} == 0
        for: 1m

[ Read 12 lines ]
^G Help      ^O Write Out  ^W Where Is   ^R Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^/ Go To Line  M-E Redo
```

Step 55: Go to /etc/alertmanager/ alertmanager.yml and make changes in the file

Step 56: add your email to the list to receive emails when there are alerts, and go to google account app password to allow prometheus to send them



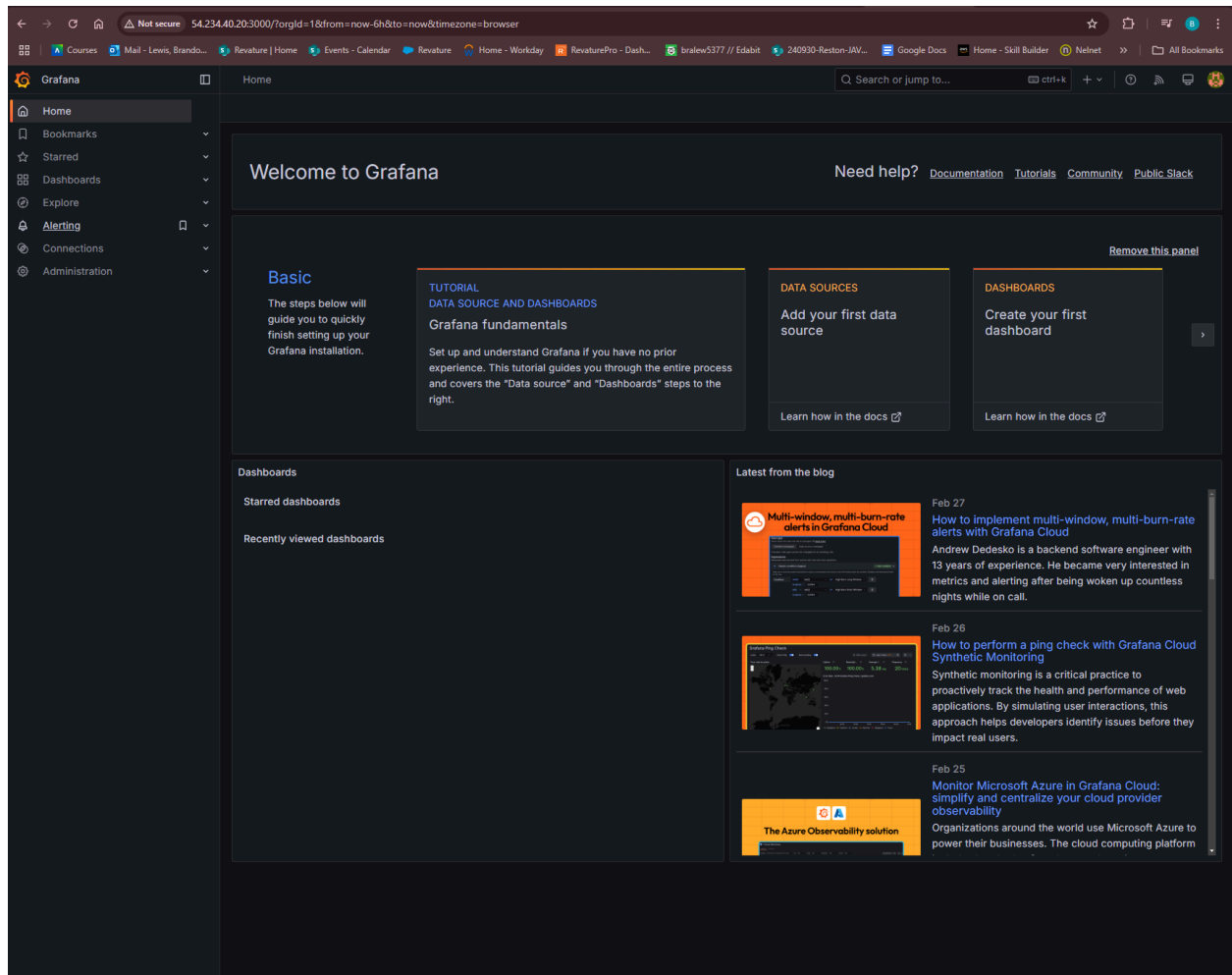
Step 57: Apt update in controller to get ready for grafana. apt update

Step 58 Follow grafana instructions

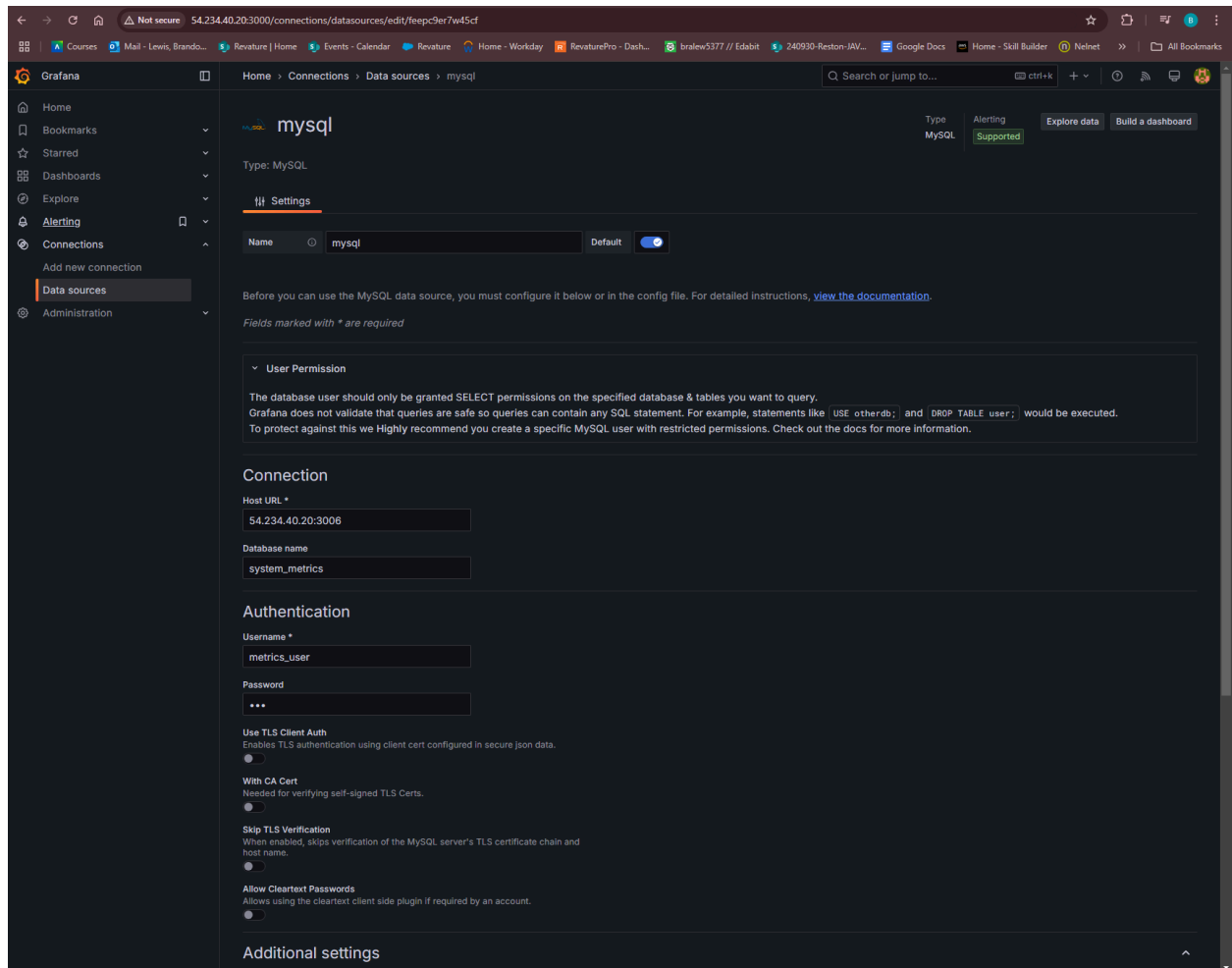
```

:apt-get install -y apt-transport-https
apt-get install -y software-properties-common wget
wget -q -O - https://packages.grafana.com/gpg.key | apt-key add -
echo "deb https://packages.grafana.com/oss/deb stable main" | tee -a
/etc/apt/sources.list.d/grafana.list
Apt update
apt install grafana -y
systemctl enable grafana-server
systemctl start grafana-server
  
```

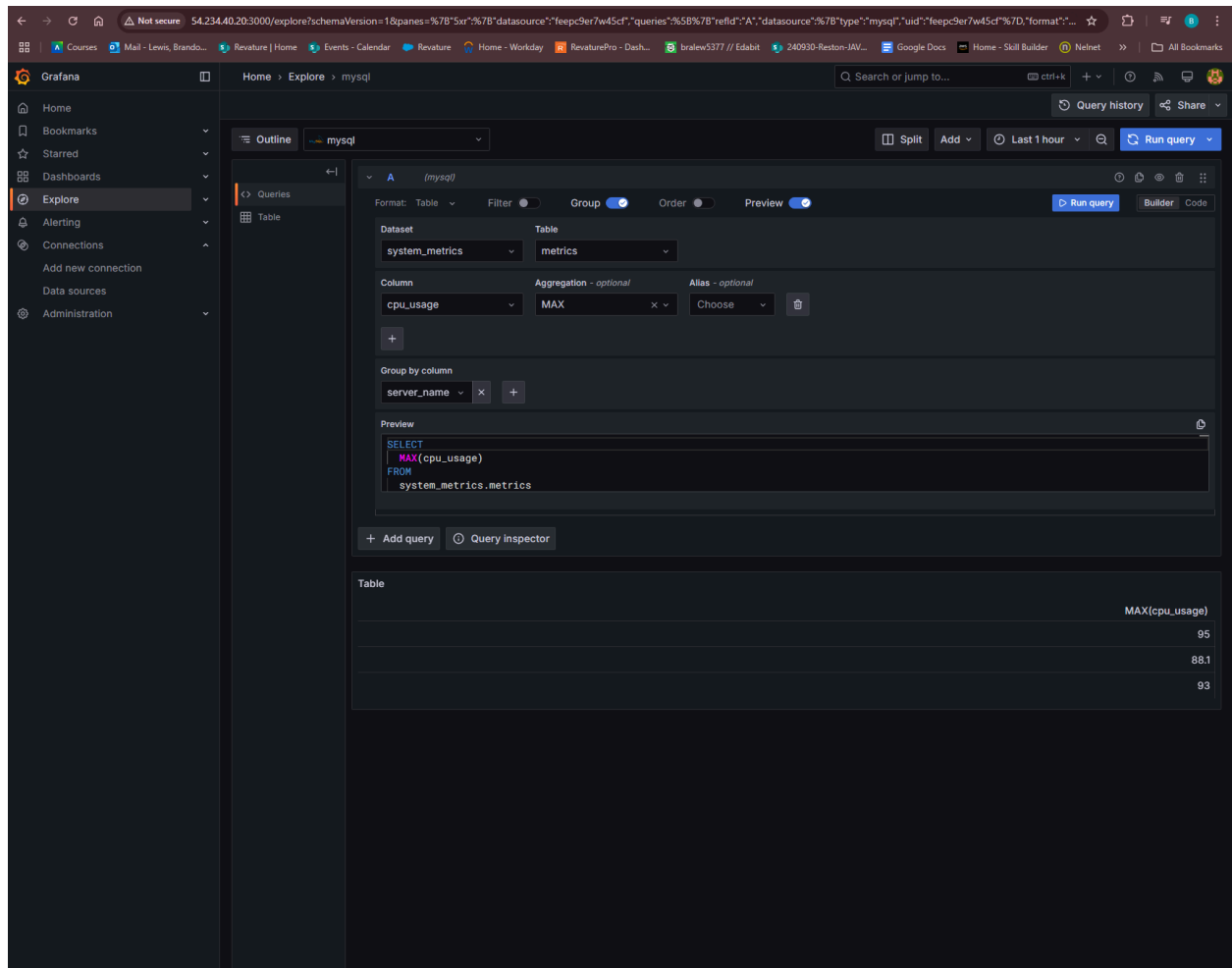
Step 59: Go to controller ip with :3000 in web browser login with admin admin



Step 60: login to database with grafana metric user and password

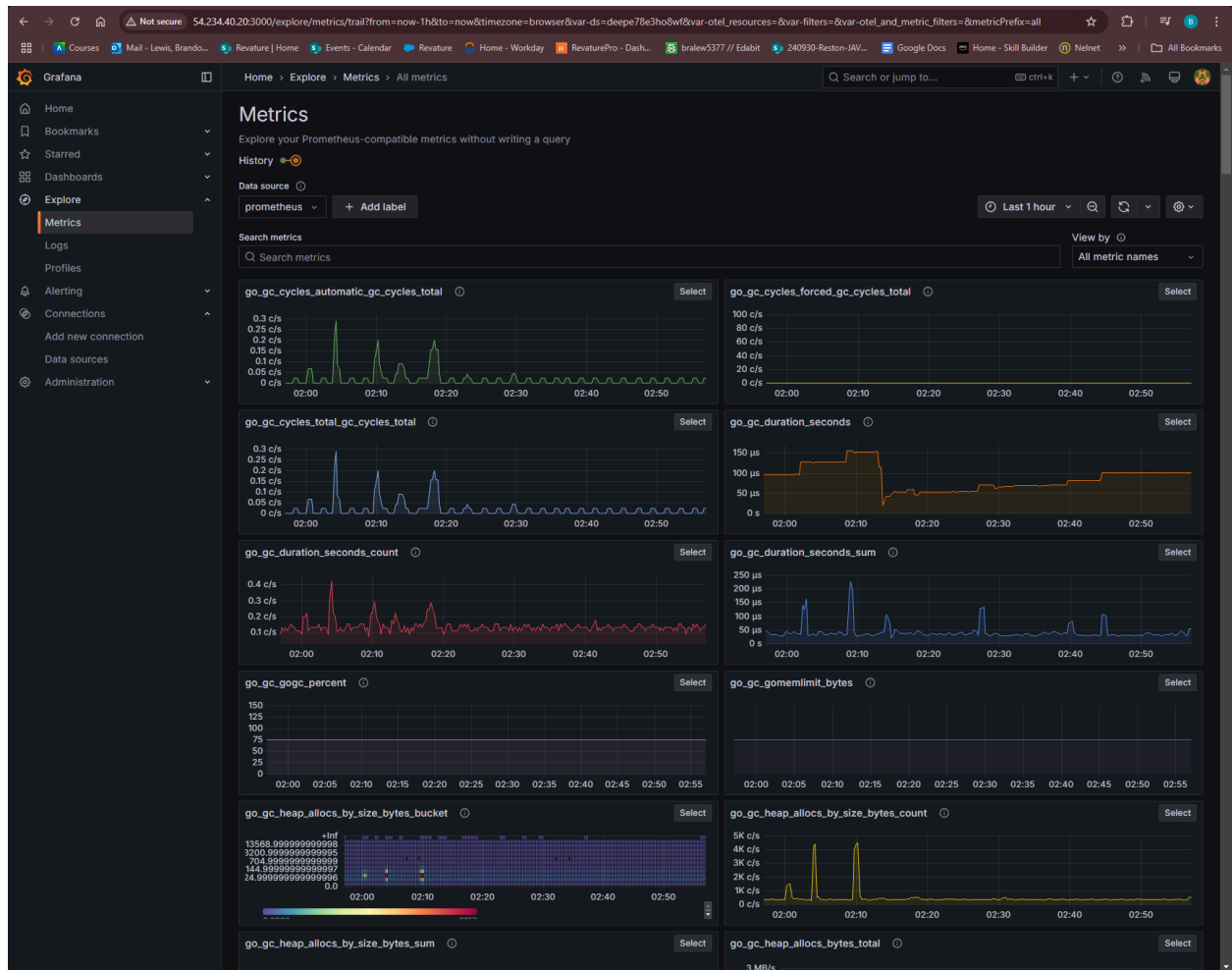


Step 61: Go to explore page and run queries to see what's happening in the database

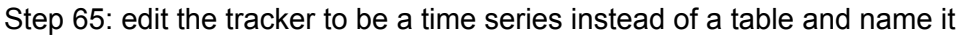


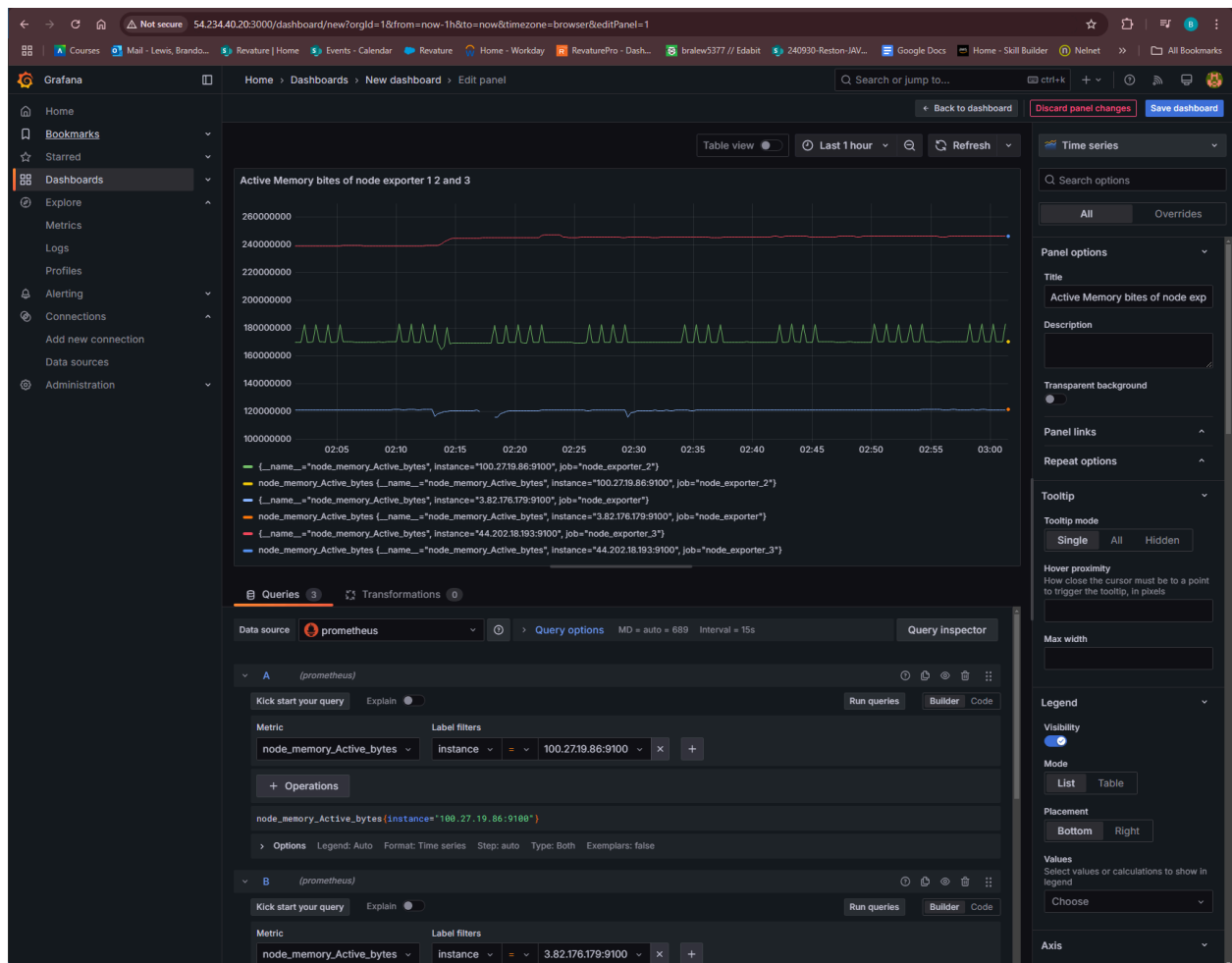
Step 62: add to dashboard

Step 63: Go to connections and add prometheus and the ip of controller with :9090 at the end



Step 64: Then go to explore and add prometheus to the tracker and then click on what you would like to track and add them for each machine and add it to the dashboard





Step 66: We now have the metrics of our machines tracked and imported to a mysql database and we can see what's happening in grafana using prometheus and node exporter !!!