

## **Guided Lab: Setting Amazon Time Sync Service for Amazon EC2 Linux Instance**

### **Description**

Accurate timekeeping is crucial for the proper functioning of many applications and services running on cloud infrastructure. This includes logging events, coordinating distributed processes, and ensuring the security of communications through protocols that rely on synchronized time, such as TLS/SSL. In the AWS ecosystem, ensuring your EC2 instances have the correct time is vital for seamless operation and accurate billing.

This guided lab will walk you through the process of configuring and managing time synchronization on a Linux instance running in Amazon EC2. You will learn how to use the Network Time Protocol (NTP) with Chrony, a versatile and powerful time synchronization tool, which is the default for many modern Linux distributions. By following these steps, you will ensure your EC2 instance maintains accurate time, improving the reliability and accuracy of time-dependent processes and logs.

### **Prerequisites**

This lab assumes you have basic knowledge of Linux command-line operations and Amazon EC2 service.

If you find any gaps in your knowledge, consider taking the following lab:

- How to launch an Amazon EC2 Linux instance

### **Objectives**

In this lab, you will:

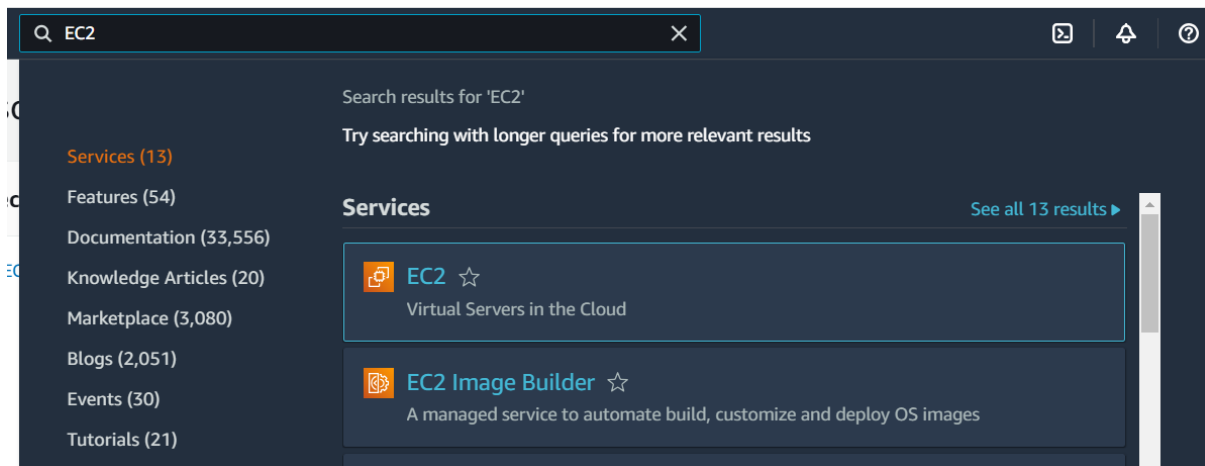
- Verify and configure time synchronization on a Linux EC2 instance.
- Understand the use of NTP and Chrony to maintain accurate time.
- Ensure your instance's time is correctly synchronized with time servers.

[Subscribe to access AWS PlayCloud Labs](#)

### **Lab Steps**

#### **Launch an EC2 Instance**

1. **Navigate the EC2 Dashboard.**



## 2. Launch an EC2 Instances using the following configurations:

- Name: **MyWebServer**
- AMI: **Amazon Linux**
- Instance type: **t2.micro**
- Key pair: (**Please create a new one.**)
  - Key pair name: **my-key-pair**
  - Key pair type: **RSA**
  - Private key file format: **.pem**
- Network settings:
  - Auto-assign public IP: Select **Enable**
  - Firewall (security groups): tick on the **Create security group**
  - Ensure that **Allow SSH traffic from** is **checked** and is **My IP**

### ▼ Key pair (login) [Info](#)

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*

my-key-pair ▼

↻ [Create new key pair](#)

### ▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-092f3b7dd98fe3bee

Subnet [Info](#)

subnet-0a234fc2136c9af52

Auto-assign public IP [Info](#)

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

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ **Allow SSH traffic from**  
Helps you connect to your instance

My IP  
49.150.103.249/32 ▼

☐ **Allow HTTPS traffic from the internet**  
To set up an endpoint, for example when creating a web server

☐ **Allow HTTP traffic from the internet**  
To set up an endpoint, for example when creating a web server

- Click on **Launch instance**
- **Verify Current Time and Time Zone**
- 1. SSH into your newly created EC2 instance.

EC2 > Instances > i-01e2e2a5493fb7e05 > Connect to instance

## Connect to instance [Info](#)

Connect to your instance i-01e2e2a5493fb7e05 (MyWebServer) using any of these options

EC2 Instance Connect	Session Manager	SSH client	EC2 serial console
----------------------	-----------------	------------	--------------------

Instance ID  
i-01e2e2a5493fb7e05 (MyWebServer)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is my-key-pair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  

```
chmod 400 "my-key-pair.pem"
```
4. Connect to your instance using its Public DNS:  

```
ec2-34-203-232-123.compute-1.amazonaws.com
```

Command copied

ssh -i "my-key-pair.pem" ec2-user@ec2-34-203-232-123.compute-1.amazonaws.com

**Note:** In most cases, if the AMI owner has provided a public key, you can connect to the instance using the public key.

- 
- 2. Check the current system time and time zone, using the commands:
- date
- timedatectl

```

~/my/
[ec2-user@ip-192-168-5-20 ~]$ date
timedatectl
Thu Jul 11 06:53:40 UTC 2024
        Local time: Thu 2024-07-11 06:53:40 UTC
        Universal time: Thu 2024-07-11 06:53:40 UTC
        RTC time: Thu 2024-07-11 06:53:40
        Time zone: n/a (UTC, +0000)
System clock synchronized: yes
        NTP service: active
        RTC in local TZ: no
[ec2-user@ip-192-168-5-20 ~]$

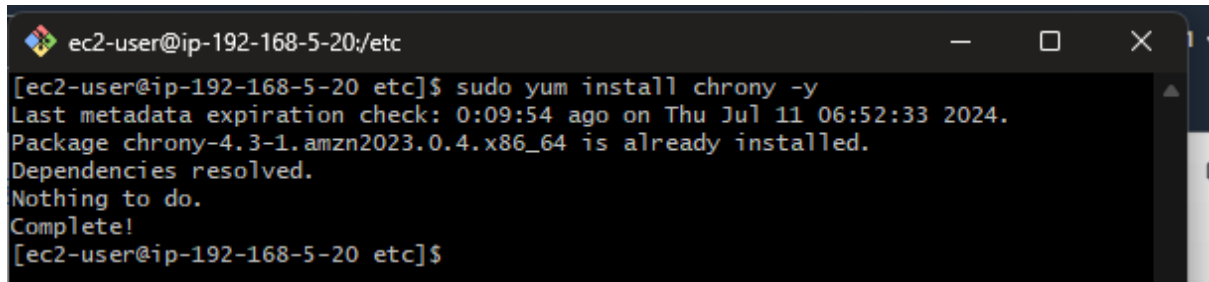
```

- 
- Take note these date and time details, and lets compare what's the difference later on.

## Install and Configure Chrony

1. Ensure that Chrony is installed on your instance. If not, install it using the package manager, using the command:

```
sudo yum install chrony -y
```

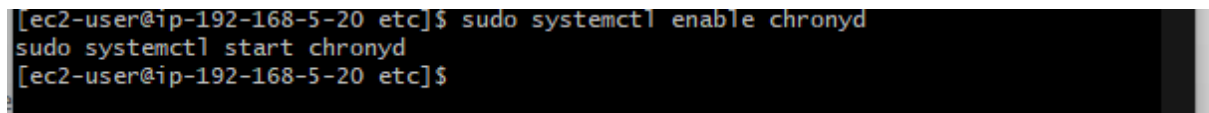


```
ec2-user@ip-192-168-5-20/etc
[ec2-user@ip-192-168-5-20 etc]$ sudo yum install chrony -y
Last metadata expiration check: 0:09:54 ago on Thu Jul 11 06:52:33 2024.
Package chrony-4.3-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-192-168-5-20 etc]$
```

2. Enable and start the Chrony service:

```
sudo systemctl enable chronyd
```

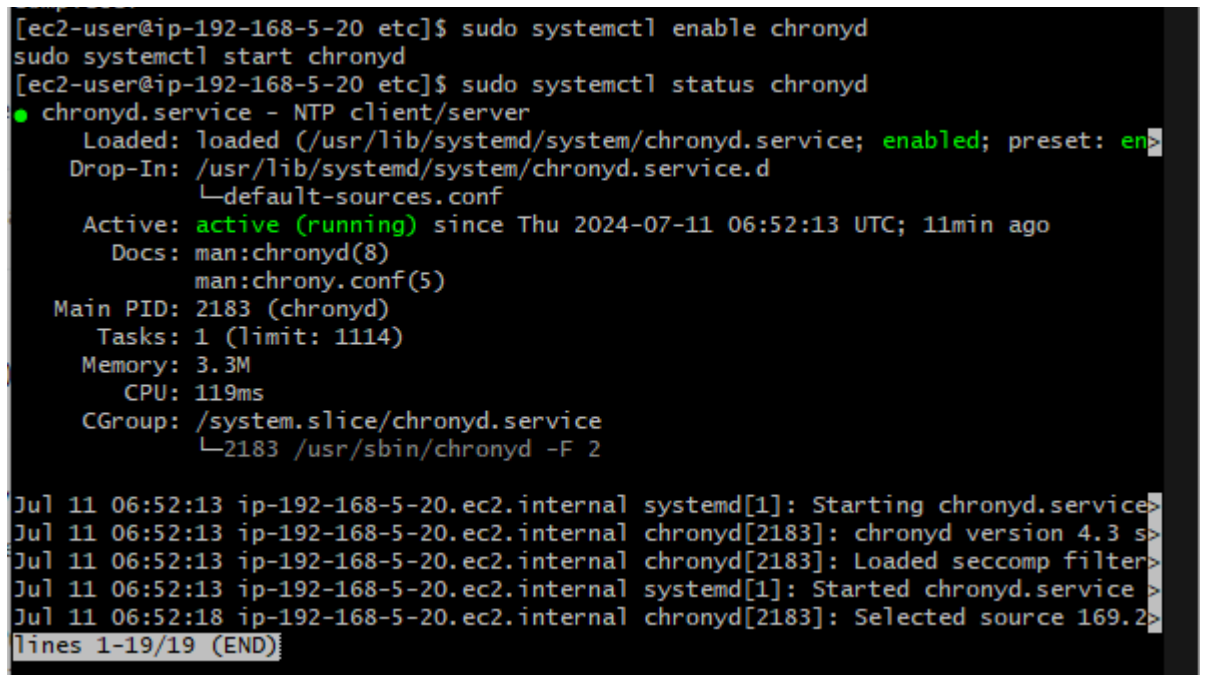
```
sudo systemctl start chronyd
```



```
[ec2-user@ip-192-168-5-20 etc]$ sudo systemctl enable chronyd
sudo systemctl start chronyd
[ec2-user@ip-192-168-5-20 etc]$
```

3. Verify the Chrony service status:

```
sudo systemctl status chronyd
```



```
[ec2-user@ip-192-168-5-20 etc]$ sudo systemctl enable chronyd
sudo systemctl start chronyd
[ec2-user@ip-192-168-5-20 etc]$ sudo systemctl status chronyd
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; preset: en>
   Drop-In: /usr/lib/systemd/system/chronyd.service.d
            └─default-sources.conf
   Active: active (running) since Thu 2024-07-11 06:52:13 UTC; 11min ago
     Docs: man:chronyd(8)
            man:chrony.conf(5)
  Main PID: 2183 (chronyd)
    Tasks: 1 (limit: 1114)
   Memory: 3.3M
      CPU: 119ms
   CGroup: /system.slice/chronyd.service
            └─2183 /usr/sbin/chronyd -F 2

Jul 11 06:52:13 ip-192-168-5-20.ec2.internal systemd[1]: Starting chronyd.service>
Jul 11 06:52:13 ip-192-168-5-20.ec2.internal chronyd[2183]: chronyd version 4.3 s>
Jul 11 06:52:13 ip-192-168-5-20.ec2.internal chronyd[2183]: Loaded seccomp filter>
Jul 11 06:52:13 ip-192-168-5-20.ec2.internal systemd[1]: Started chronyd.service>
Jul 11 06:52:18 ip-192-168-5-20.ec2.internal chronyd[2183]: Selected source 169.2>
lines 1-19/19 (END)
```

4.(OPTIONAL) Configure Chrony by editing the configuration file /etc/chrony.conf (optional, if you need to change the default NTP servers or settings):

```
sudo vi /etc/chrony.conf
```

5. Verify Time Synchronization by checking the Chrony tracking and sources:

chronyc sources -v

```
[ec2-user@ip-192-168-5-20 ~]$ chronyc tracking
chronyc sources -v
Reference ID      : A9FEA97B (169.254.169.123)
Stratum          : 4
Ref time (UTC)   : Thu Jul 11 07:12:31 2024
System time      : 0.000000332 seconds fast of NTP time
Last offset      : -0.000000510 seconds
RMS offset       : 0.000005419 seconds
Frequency        : 26.628 ppm slow
Residual freq    : -0.000 ppm
Skew             : 0.025 ppm
Root delay       : 0.000516500 seconds
Root dispersion  : 0.000042239 seconds
Update interval  : 16.3 seconds
Leap status      : Normal

.-- Source mode  '^' = server, '=' = peer, '#' = local clock.
/ .- Source state '*' = current best, '+' = combined, '-' = not combined,
| /              'x' = may be in error, '~' = too variable, '?' = unusable.
||
||              .- xxxx [ yyyy ] +/- zzzz
||      Reachability register (octal) -.      | xxxx = adjusted offset,
||      Log2(Polling interval) --.      |      | yyyy = measured offset,
||              \            |      |      | zzzz = estimated error.
||              |            |      |
MS Name/IP address         Stratum Poll Reach LastRx Last sample
=====
A* 169.254.169.123         3    4   377     4  +2200ns[+1690ns] +/-  339us
A- ec2-54-197-201-248.compu> 4    6   377    48   -25us[ -24us] +/- 1080us
A- ec2-3-87-127-143.compute> 4    7   377   115   -33us[ -33us] +/-  878us
A- ec2-54-90-191-9.compute-> 4    7   377   115   -36us[ -36us] +/-  674us
A- ec2-3-86-4-106.compute-1> 4    7   377   118  -8910ns[-9732ns] +/-  601us
[ec2-user@ip-192-168-5-20 ~]$
```

## Set the Correct Time Zone

1. List available time zones:

```
timedatectl list-timezones
```

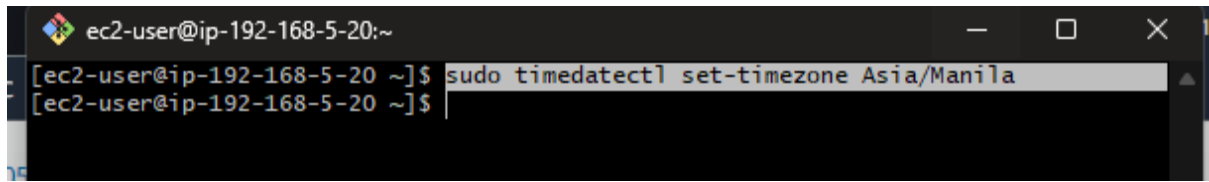
```
ec2-user@ip-192-168-5-20:~  
[ec2-user@ip-192-168-5-20 ~]$ timedatectl list-timezones  
Africa/Abidjan  
Africa/Accra  
Africa/Addis_Ababa  
Africa/Algiers  
Africa/Asmara  
Africa/Asmera  
Africa/Bamako  
Africa/Bangui  
Africa/Banjul  
Africa/Bissau  
Africa/Blantyre  
Africa/Brazzaville  
Africa/Bujumbura  
Africa/Cairo  
Africa/Casablanca  
Africa/Ceuta  
Africa/Conakry  
Africa/Dakar  
Africa/Dar_es_Salaam  
Africa/Djibouti  
Africa/Douala  
Africa/El_Aaiun  
Africa/Freetown  
Africa/Gaborone  
Africa/Harare  
Africa/Johannesburg  
Africa/Juba  
Africa/Kampala  
Africa/Khartoum  
Africa/Kigali  
Africa/Kinshasa  
Africa/Lagos  
Africa/Libreville  
Africa/Lome  
Africa/Luanda  
Africa/Lubumbashi  
Africa/Lusaka  
Africa/Malabo  
Africa/Maputo  
Africa/Maseru  
Africa/Mbabane  
Africa/Mogadishu  
Africa/Monrovia  
Africa/Nairobi  
Africa/Ndjamena  
Africa/Niamey  
Africa/Nouakchott  
Africa/Ouagadougou  
Africa/Porto-Novo  
Africa/Sao_Tome  
Africa/Timbuktu  
Africa/Tripoli  
Africa/Tunis  
Africa/Windhoek  
America/Adak  
America/Anchorage  
America/Anguilla
```

Hit ENTER to check the check the other timezones or next page

Click on CTRL + C to exit

2. Set your desired time zone (e.g., Asia/Manila):

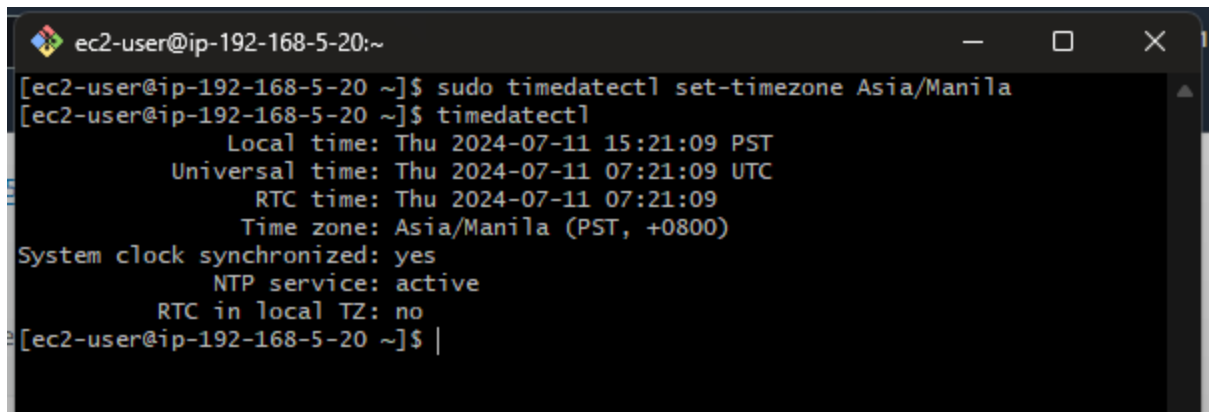
```
sudo timedatectl set-timezone Asia/Manila
```

A terminal window with a dark background. The prompt is 'ec2-user@ip-192-168-5-20:~'. The command 'sudo timedatectl set-timezone Asia/Manila' has been entered and is highlighted in a light blue selection box. The cursor is at the end of the command line.

```
ec2-user@ip-192-168-5-20:~  
[ec2-user@ip-192-168-5-20 ~]$ sudo timedatectl set-timezone Asia/Manila  
[ec2-user@ip-192-168-5-20 ~]$
```

3. Verify the changes:

```
timedatectl
```

A terminal window showing the output of the 'timedatectl' command. The output displays local, universal, and RTC times, the current time zone (Asia/Manila), and the status of the system clock and NTP service.

```
ec2-user@ip-192-168-5-20:~  
[ec2-user@ip-192-168-5-20 ~]$ sudo timedatectl set-timezone Asia/Manila  
[ec2-user@ip-192-168-5-20 ~]$ timedatectl  
          Local time: Thu 2024-07-11 15:21:09 PST  
        Universal time: Thu 2024-07-11 07:21:09 UTC  
             RTC time: Thu 2024-07-11 07:21:09  
           Time zone: Asia/Manila (PST, +0800)  
System clock synchronized: yes  
             NTP service: active  
      RTC in local TZ: no  
[ec2-user@ip-192-168-5-20 ~]$
```

That's it! You have successfully configured time synchronization on your Linux EC2 instance using Chrony. Ensuring accurate time synchronization is essential for maintaining system logs, data consistency, and secure communications. You have also learned how to set the correct time zone for your instance. Regularly verifying and managing time settings is a good practice to ensure the smooth operation of your applications and services.

Accurate timekeeping helps in diagnosing issues by providing reliable timestamps in logs, facilitates seamless coordination across distributed systems, and ensures the integrity and security of communications and transactions. Regularly verifying and managing time settings is a good practice to ensure the smooth operation of your applications and services.

By mastering these skills, you can enhance the reliability and performance of your cloud infrastructure, making sure your systems are always operating with precise time synchronization. Continue to apply these practices in your future deployments and maintenance tasks to maintain optimal system health. Happy Learning!