**Terraform**

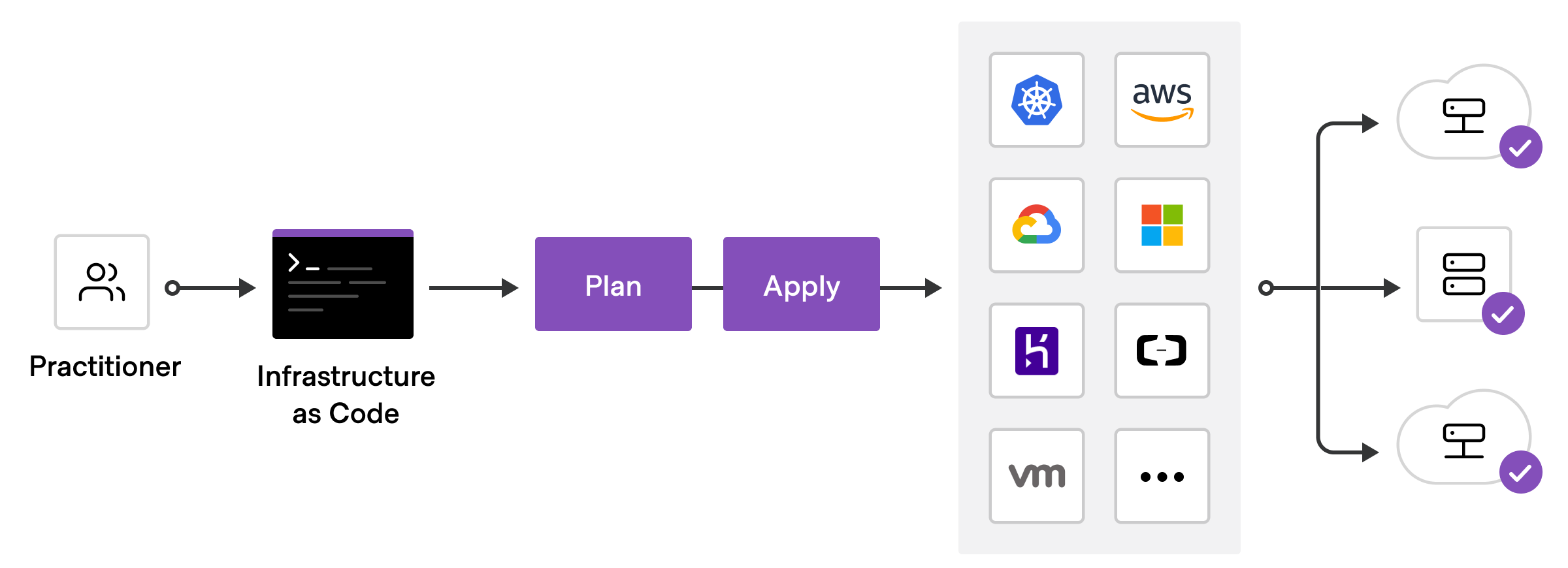
* Terraform by HashiCorp, an AWS Partner Network (APN) Advanced Technology Partner and member of the AWS DevOps Competency, is an “infrastructure as code” tool similar to AWS CloudFormation that allows you to create, update, and version your Amazon Web Services (AWS) infrastructure.
* Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently. Terraform can manage existing and popular service providers as well as custom in-house solutions.
* Terraform generates an execution plan describing what it will do to reach the desired state, and then executes it to build the described infrastructure

**Why Terraform?**

* Reduce human error and increase automation by provisioning infrastructure as code.
* Provision infrastructure across 300+ public clouds and services using a single workflow.
* Provision consistent testing, staging, and production environments with the same configuration.

**How Terraform Works?**

* Terraform allows infrastructure to be expressed as code in a simple, human readable language called HCL (HashiCorp Configuration Language).
* It reads configuration files and provides an execution plan of changes, which can be reviewed for safety and then applied and provisioned.
* Extensible providers allow Terraform to manage a broad range of resources, including IaaS, PaaS, SaaS, and hardware services.



**Using Terraform has several advantages over manually managing your infrastructure:**

* Terraform can manage infrastructure on multiple cloud platforms.
* The human-readable configuration language helps you write infrastructure code quickly.
* Terraform's state allows you to track resource changes throughout your deployments.
* You can commit your configurations to version control to safely collaborate on infrastructure.

**Terraform has a great set of features that make it worth adding to your tool belt, including:**

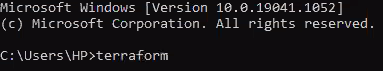
* Friendly custom syntax, but also has support for JSON.
* Visibility into changes before they actually happen.
* Built-in [graphing](https://www.terraform.io/docs/commands/graph.html) feature to visualize the infrastructure.
* Understands resource relationships. One example is failures are isolated to dependent resources while non-dependent resources still get created, updated, or destroyed.
* [Open source project](https://github.com/hashicorp/terraform) with a community of thousands of contributors who add features and updates.
* The ability to break down the configuration into smaller chunks for better organization, re-use, and maintainability. The last part of this article goes into this feature in detail.

**To deploy infrastructure with Terraform:**

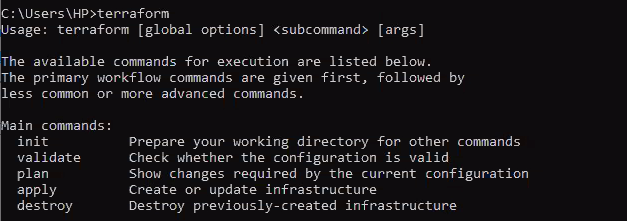
* **Scope** - Identify the infrastructure for your project.
* **Author** - Write the configuration for your infrastructure.
* **Initialize** - Install the plugins Terraform needs to manage the infrastructure.
* **Plan** - Preview the changes Terraform will make to match your configuration.
* **Apply** - Make the planned changes.

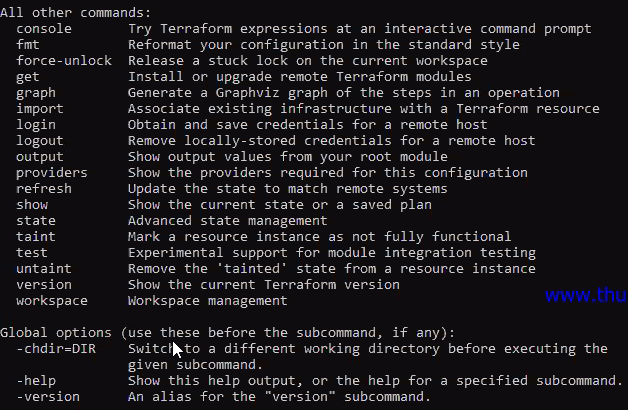
**How to Configure Terraform on Windows?**

* Visit <https://www.terraform.io/downloads.html>
* Download and Install Terraform by unzipping it and moving it to a directory included in your system's path.
* Copy the path where downloaded file is stored: E:\DevOps\Terraform
* Right click Computer, click on Properties and select Advanced settings
* Select option “Environment Variables”
* Under System variable section, select path and click on Edit
* In Edit environmental variable, select New and paste copied path E:\DevOps\Terraform and click on OK
* Now go command prompt and check



**Please note down below commands:**

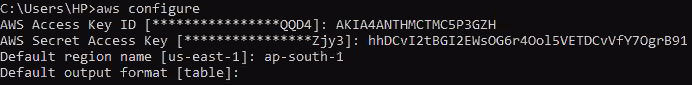




**How to establish connection between Terraform and AWS and create EC2 Instance?**

* Go to you AWS account => IAM Services
* Click on “Users” available under IAM resources
* Click on Add User
* Create Username as Terraform
* Under Select AWS access type, choose “Programmatic Access”.
* Click on Next: Permissions
* Click on “Attach existing policies directly” to assign existing policies.
* Search for “AdministratorAccess ”, select the same and click on Next: Tags
* Click on Next: Review
* Click on Create User
* Once User is created successfully, Access Key ID and Secret Access Key is generated
* You may download the same by clicking on “Show”
* Now we have to establish connection between our local work machine i.e. Terraform and AWS with the help of Username.
* Go to Command Prompt, invoke a command: aws configure and provide below mentioned details:

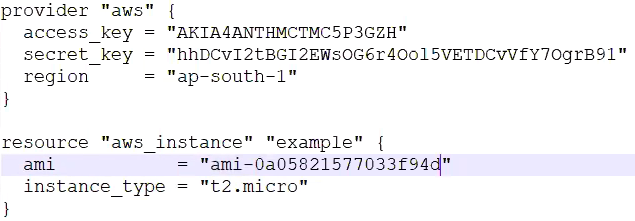
1. Access Key ID
2. Secret Access Key
3. Region Name



* To download all Terraform plug-ins you need to pass a command: terraform init
* But to execute the same, it looks for terraform configuration file where credential details and information regarding which/what task to be executed will be mentioned.
* Open Note pad and write program to be executed (to create AWS instance)
* Save it as instance.tf format under E:\DevOps\Terraform directory

**Note**:

* Select “Save as type” – All types
* AMI ID specified in program is of available Amazon Machine Images not of EC2 Instance.
* Ensure you choose (64-bit x86) AMI ID.



* Go pack to command prompt
* Go to directory E:\DevOps\Terraform

**Command**:

**C:\Windows \System32> E:**

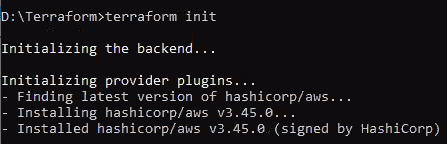
**E:\>cd DevOps**

**E:\DevOps> cd Terraform**

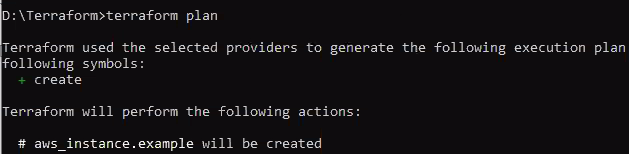
* To view files located under directory E:\DevOps\Terraform pass below command

**E:\DevOps\Terraform>dir**

* Once you invoke “terraform init” command, it looks for terraform configuration file (.tf file)
* Since we have mentioned provider as “AWS” in the program, it will connect to AWS and download all plugins required to interact terraform with AWS
* Invoke a command: **terraform init**



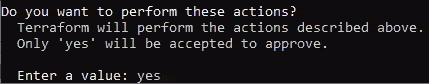
* Invoke **terraform plan**
* The terraform plan command creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure.



* Pass **terraform apply** command. The terraform apply command executes the actions proposed in a Terraform plan.



* Enter value as “Yes”



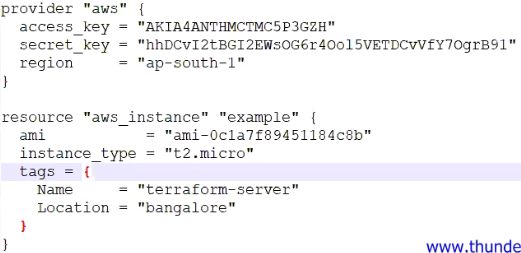
* Once you enter the value, actions proposed in a Terraform plan will be executed (Create EC2 Instance)
* EC2 instance will be created
* Now go to the directory E:\DevOps\Terraform and if you observe .tfstate is created.

**Note:**

* Whenever you create a resource, a file called .tfstate will be created.
* All the resource we create using terraform script is maintained in a .tfstate file which is in a jason format.
* Default name would be Terraform.tfstate i.e. your local work space
* Ensure .tfstate file is not modified/deleted
* In case you modify/delete .tfstate file, there will be inconsistency in your environments.
* Hence it is advised to store .tfstate file in Amazon S3 storage rather than in your local work space, then apply lock. And only one user to be allowed to have access to the file
* So that we can avoid other users trying to modify/delete the file.

**How to add tag?**

* Open instance.tf file and edit the program as mentioned in below screenshot.



* Invoke **terraform plan**



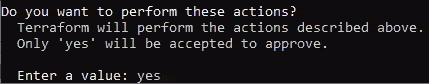
* Since we have just modified in the same existing file, it is indicating “1 to change” as you observe in below output



* Pass a command: terraform apply



* Enter value as “Yes”



* Go to AWS=> EC2 Instance and check whether tag (terraform-server) is updated or not.

**How to destroy/terminate EC2 instance?**

* Invoke command : terraform destroy



* Go to AWS=> EC2 Instance and check whether instance, terraform-server is deleted or not.

**Note:** To gain more knowledge on Terraform please do visit below mentioned link

1. **YouTube** : <https://www.youtube.com/playlist?list=PLH1ul2iNXl7vk8RUchIiMBeXqDnFTi4_M>
2. **Website**: <https://www.terraform.io/>
3. **Website**: <https://learn.hashicorp.com/collections/terraform/aws-get-started>

**Continuous Monitoring**

**Definition**: Continuous Monitoring (CM), sometimes called Continuous Control Monitoring (CCM), is process of overseeing the entire development process from planning, development, integration and testing, deployment, and operations.

**How** **Continuous Monitoring works?**

* Once the software is released into production, Continuous Monitoring will notify dev and QA teams in the event of specific issues arising in the prod environment.
* It provides feedback on what is going wrong, which allows the relevant people to work on necessary fixes as soon as possible.
* Continuous Monitoring basically assists IT organizations, DevOps teams in particular, with procuring real-time data from public and hybrid environments.
* This is especially helpful with implementing and fortifying various security measures – incident response, threat assessment, computers, and database forensics, and root cause analysis.
* It also helps provide general feedback on the overall health of the IT setup, including offsite networks and deployed software.

**Types of Continuous Monitoring:**

* **Infrastructure Monitoring**: Monitors and manages the IT infrastructure required to deliver products and services. This includes data centers, networks, hardware, software, servers, storage, and the like. Infrastructure Monitoring collates and examines data from the IT ecosystem to improve product performance as far as possible.
* **Application Monitoring**: Monitors the performance of released software based on metrics like uptime, transaction time and volume, system responses, API responses, and general stability of the back-end and front-end.
* **Network Monitoring:** Monitors and tracks network activity, including the status and functioning of firewalls, routers, switches, servers, Virtual Machines, etc. Network Monitoring detects possible and present issues and alerts the relevant personnel. Its primary goal is to prevent network downtime and crashes.

**Benefits of Continuous Monitoring:**

* **Better Network Visibility and Transparency:** CM offers DevOps teams clarity on the state of the IT infrastructure by automatically collecting and analyzing data to reflect possible outages and important trends.
* **Facilitates Rapid Responses:** A primary aspect of CM is implementing an alert system that immediately notifies the right people the minute an IT incident emerges. This enables timely response to security threats or functional stop-gaps, minimizing damage and allowing faster restoration of the system to optimal operational levels.
* **Minimizes System Downtime:** Consistent system monitoring and quick, necessary alerts help maintain system uptime by raising the alarm when there is a service outage or any application performance issues.
* **Assists with Healthy Business Performance:** Reduction in system downtime also minimizes negative impact on customer experience, thus safeguarding the organization against losses in revenue or credibility. As mentioned before, Continuous Monitoring tools can also be used to track user reactions to software updates, which is useful for several teams – development, QA, sales, marketing, customer service, etc.

**Logging & Log file:**

* A log file is a computer-generated data file that contains information about usage patterns, activities, events and operations within an operating system, application, server or another device.
* The process of capturing data/events into log file for later retrieval and analysis is called logging.
* Most Linux log files are stored in a plain ASCII text file and are in the**/var/log** directory and subdirectory. Logs are generated by the Linux system daemon log, **syslogd** or **rsyslogd**.

Mainly there are four types of log files generated in a Linux based environment and they are:

* Application Logs.
* Event Logs.
* Service Logs.
* System Logs.

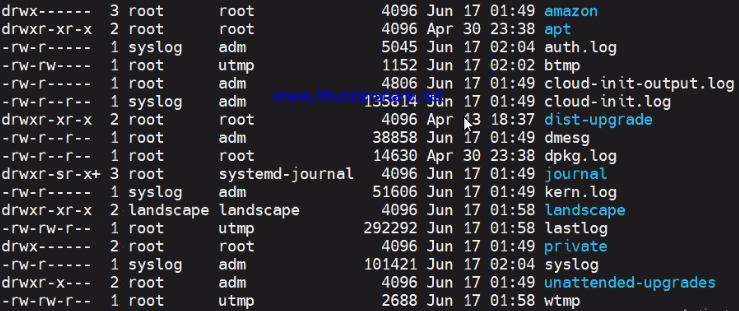
**What are the uses of Linux log files?**

* From the log files, one can observe and find the details on server performance, security, error messages, and underlying issues.
* Any issue that the server is undergoing one can get the clue by detailed view on the log files.
* Hence by reviewing the log files one can solve the existing issues and can take precautions for the issues that may cause in future!

**How to track log files?**

* Create EC2 Instance with Ubuntu Image
* Open MobaXterm
* Create new SSH session with public IP address of EC2 Instance
* Login as admin (sudo)
* Go to /var/log directory and list out files contained it.





* You may list out (ls -l) each log files and check the content of it.

Note that log files are stored in plain text so they can be viewed by using the following standard commands:

* **zcat**– Displays all the contents of **logfile.gz**
* **zmore**– See the file in pages, without decompressing the files
* **zgrep**– Search inside a compressed file
* **grep**– Find all occurrences of a search term in a file or filter a log file
* **tail**– Output the last few lines of files
* **head**– View the very beginning of text files

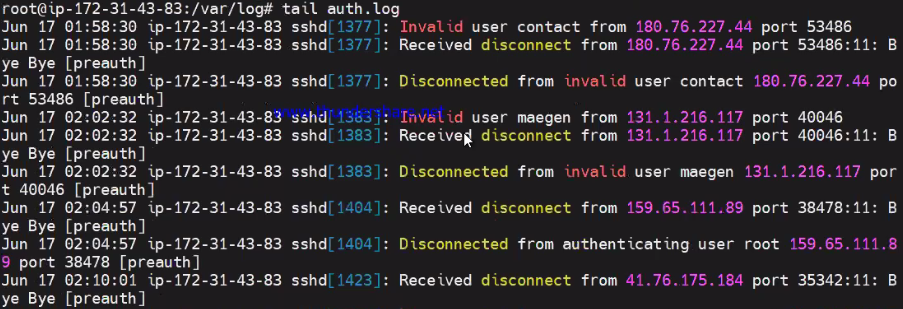
Let’s see how tail command functions:

* Instead of listing out all logs, you may list out only the last 10 lines of file; which can be achieved by

**Syntax**: tail (File Name)

**Command**: tail auth.log

* As you can observe in screenshot (Refer Next Page), only few lines of auth.log file are captured.



* In case you want you want to check how logs are captured, open duplicate SSH session of your EC2 instance.
* Perform any activity over there and same thing will be captured under /var/log

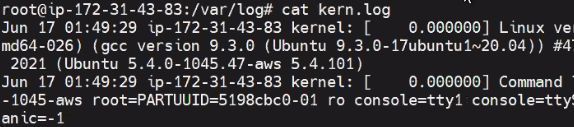


* Now go back to original session and pass command tail auth.log
* As you can observe below, activity has been captured into auth.log file



* In case, you want to fetch kernel related logs, pass below command.

**Command**: cat kern.log



**What is logrotate?**

* logrotate is designed to ease administration of systems that generate large numbers of log files.
* It allows automatic rotation, compression, removal, and mailing of log files.
* Each log file may be handled daily, weekly, monthly, or when it grows too large.
* The logrotate program is a log file manager. It is used to regularly cycle (or rotate) log files by removing the oldest ones from your system and creating new log files.
* It may be used to rotate based on the age of the file or the file's size, and usually runs automatically through the cron utility
* To verify if a particular log is indeed rotating or not and to check the last date and time of its rotation, check the /var/lib/logrotate/status file.

**Note**: By default, there will be script by the name located at /etc/logrotate.d/rsyslog

* Go to /etc/logrotate.d/rsyslog directory and list out (ls -l) files located in it

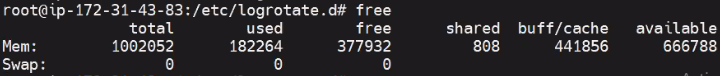




* You may list out files located in logrotate.d and open rsyslog.

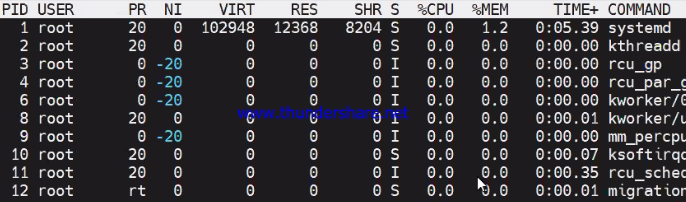


* Once you open rsyslog, you can observe the default script is written to execute automatic rotation, compression, removal, and mailing of log files on daily, weekly or monthly basis as per the requirement.
* In case you want to fetch storage related information, invoke “free” command



* To fetch information related real time tasks, processes, performance and CPU utilization, pass “top” command





* Now navigate back to root



* In case you want to find a certain file and if you are unaware of its location, specify format of file as shown below and all files will be listed out.

**Syntax**: find –name “\*.file format\*”

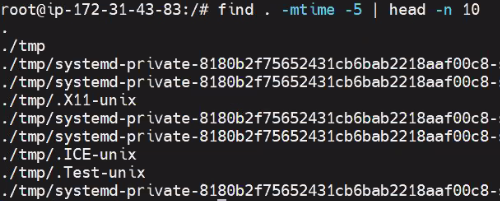
**Command**: find –name “\*.profile\*”



**Command**: find –name “\*.conf\*”

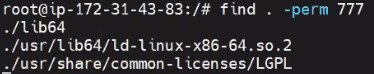


* Suppose you want to find files created/modified recently or say, in last 5 days invoke below mentioned **command**: find . –mtime -5 | head –n 10



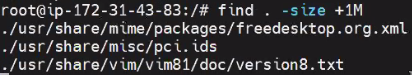
* In case you want to list out files which have been granted full permission, pass below command

**Command**: find . –perm 777



* Once the above mentioned command is executed, all the files with full permission will be listed out.
* In case you want to find file of certain size (Ex: 1 mb), invoke below command.

**Command**: find . –size +1M



* Suppose, you want to check since how long your system is functioning, load average and number of users logged in, pass “uptime” command

****

* Strace command captures and records all system calls made by a process and the signals received by the process.



* Suppose you want check output signals of process, invoke below mentioned command.

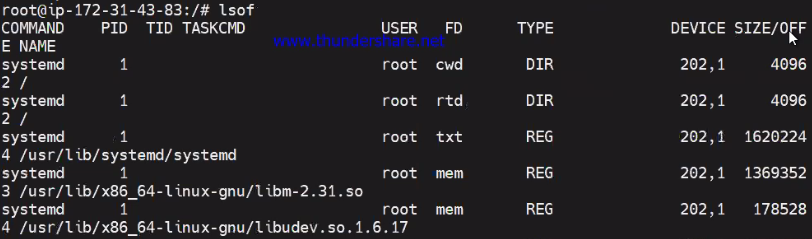
**Command**: strace –o output ls



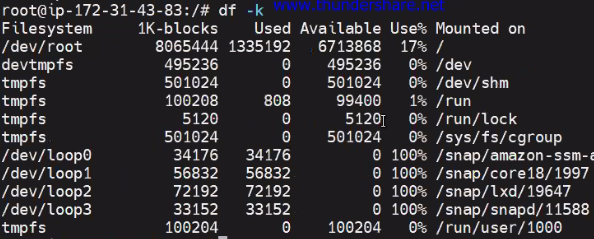
* There is something called “mount” command. The mount command mounts a storage device or filesystem, making it accessible and attaching it to an existing directory structure.



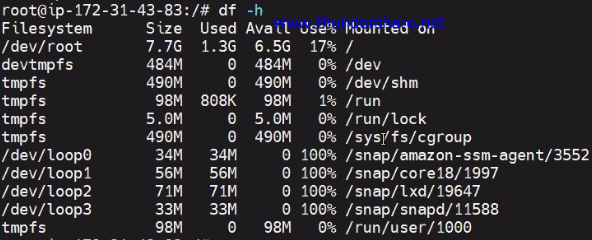
* In case you want to list out files which are open, pass lsof command.
* lsof command stands for List Of Open File. This command provides a list of files which are opened by which process.



* Suppose you want to check property details of your file system in kilo bytes, invoke a **command**: df -k

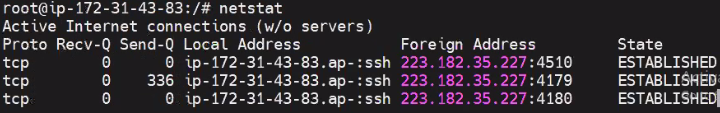


* In case you want to check property details of your file system in kilo bytes, invoke a **command**: df -k

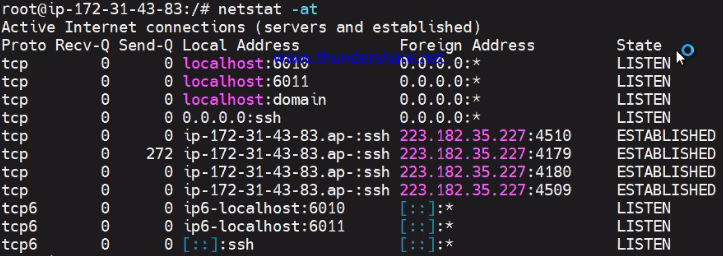


* In case you want to check network statistics, pass a **command**: netstat

**Prerequisite**: To check network status, net-tools to be installed.



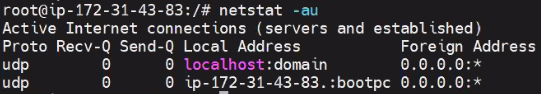
* Suppose you want to check only TCP networks (Servers and Established), invoke **command**: netstat -at



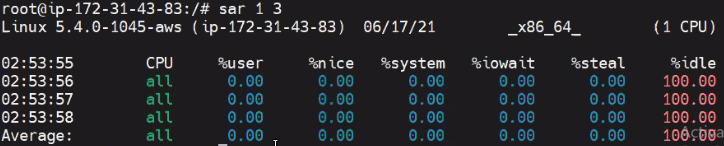
* In case you want check only active servers, invoke **command**: netstat -tl



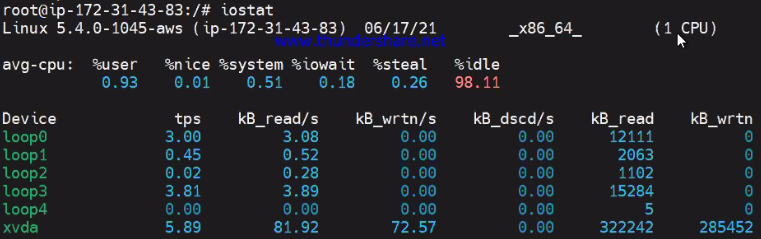
* Suppose you want to check only UDP network, invoke **command**: netstat -au



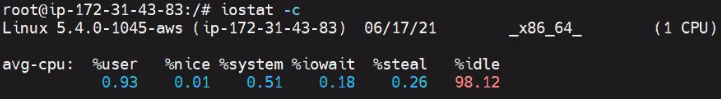
* SAR collects, report, or save system activity information. It's a utility used to collect and report system activity.



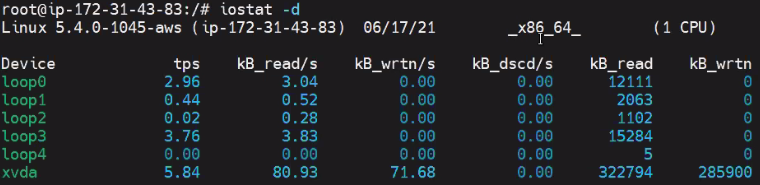
* iostat command is a command that used for monitoring system input/output device loading by observing the time the devices are active in relation to their average transfer rates



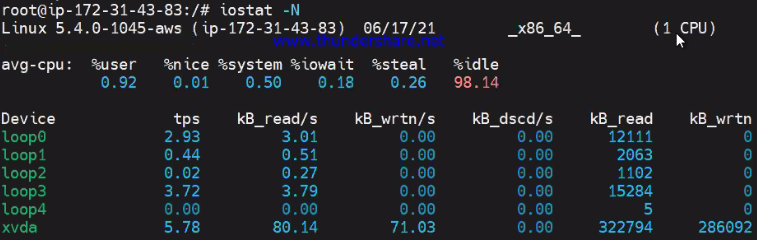
* In case you want to check CPU statistics, pass **command**: iostat -c



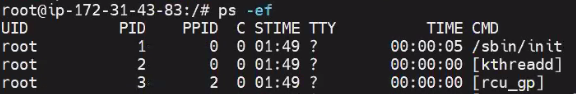
* Suppose you want to check disk I/O statistics, pass a command: iostat –d



* In case you want to check network statistics, invoke a command: iostat –N



* In case you want to check process running in your system, pass a command: ps -ef



* Suppose you want to sort out list of specific process, invoke a command: ps –ef | grep ubuntu

