Big Data

Huge data  
  
We need to work on storignand processing the data

Problems with big data

1.volumne

2.velocity

3.value

4.variety  
  
  
  
 **Data layers** Storage layer

Processing layer  
Testing layer  
Visualization laye

Data science and machine learning layer,artificial intelligence

Automation layer

Scheduling layers

**Histoy of hadoop**

**2002->Google file system was launched**

**2004->Google map reduce cam e into existnec**

2005-6>hadoop was invented by (dugh cutting who was an ex employee of yahoo but later started his own company

He brought HDFS and MapReduce into existence(It was open source)  
  
Path of dE

  
  
 **HADOOP**

Haddop has only two projects Hdfs→based of gfs and Map reduce based on google map reduce  
HIve spark scooby all are built on top of hadoop.We have to first install hadoop then install teh others  
  
Some questions:  
  
1.What is a file system?  
For each oS there is a layer between software and hardware

File system is used to read data to and from the harddisk.Ex:NTFS,S3,HDFS  
  
2. **What is a block?**   
When we need to store some 1gb of data into the hard disk then we will break the data into blocks thus data will be stored in form of blocks.The data is stored in form of blocks because it makes it easy to read and write operations.There is a fixed size of block in each OS  
  
3.**Client and server  
  
4.Types of file systems**

Standalone and distributed file system  
**  
Ntfs is** standalone from the data engineer perspective but hdfs is distributed because when some data comes then inside ntfs it will be stored on one computer only but when hdfs stores data it breaks down data in blocks and stores on different machines.  
  
5.**Types of distributed file systems  
a)Peer to peer**

**b)maater slave**

**  
1.Peer to Peer:**

The data is distributed among all the peers they talk tp each other

**2.Master slave**

This is teh most commonly used file system.HDFS follows this but it has an issue of SPOF(Single point of failure) due to SPOC(Single point of contact).The salves cannot interact with each other in this   
  
6. **What is a process?**

A program in execution is a process.

**7.what is a daemon process?** A program running inthe background is called as demon.  
8. **What is a cluster an dnode?**

Node:A single physical or virtual machine is a node.

Cluster:A group of nodes is a cluster.  
  
  
  **Block size** The block size is in multiples of 2 .  
64 mb or 218 mb  
  
 **Replication** Replication strategy is 3.  
If you want to store 1Gb then you need 3GB space since the replication factor is 3.  
The data is stored at three nodes so that a user will never know if any of the machine is down.  
  
  
 **HDFS**

**Hadoop distributed file system** A company asks you to set up a five node cluster then rather than having 5 physical computers it is better to have 5 virtual computers from amazon>  
OS:hadoop supports windows and linux.But it is better o have linux  
5 machines with linux installed  
NOw out if the 5 machines you have to decide which one do you want to make a master.Keep in mind that the master machins should have high configuration high ram and storage.Inintially these are simple machines then we need to download hadoop  
  
  
 **Downloading hadoop**

There is no such thing as different hadoop for master slave .Hadoop is same for all and can be easily downloaded.

**Daemon process**  
NOw when hadoop is downloaded then will have 5 daemon process.(jp1,jp2,jp3,jp4,jp5).out of theses first 3 will be used by hdfs. And 2 by mapreduce  
  
JP1 Process is run in master and JP2 is run in remaining 4 machines  
  
  
 **EDGE NODE** The concept of edge node is such that a developer cannot directly interact with the slave ad master node so whtaever he needs to do whether write spark job or hadoop files h will write on the edge node which will further move it to the slave node.

  
  
 **Uploading data**

When the data is uploaded to a slave node the file system is ntfs and when we move to the data to hdfs then it will become distributed.for every request of data upload a client api will be made.

  
**Client API** will send the details to the master on what type of request it is and also the name of the file. And file size.  
  
Master will generate metadata about the data now.  
Metadat will have:

1.file size

2.block size given in hdfs

3.replication(for each block how many replicas to be made)

4.placement allocation (where the data block has to sit→this is decided by rack awareness policy)-->block allocated to respective node  
This metadata will not be stored in HDFS but it will be stored in NTFS.  
Now the response will be sent to the client API to tell where to put the block on.



Client api will further make a pipeline on the node.  
Pipeline will now make blocks of the data and send to respective nodes.

**Heart Beat**

Every three seconds the slave will communicate to the master .this is known as heart beat.Based on this only the master will know that the slave is alive.

**Failure**

Failure can be due to software issue or due to software issue and network failure  
  
Suppose a node does not give a heart beat hence master will think it is dead so it will do a automatic failure process request.what it will do is it will check the block it has is it not present on some machine send it there.  
  
Now scenario can be that due to some fluctuation the data of node 3 is come again but blok b0 on it had been replicated to some other node then we will have 4 copies of b0 hence next time request comes for b0 then the request to fetch t will not go to node.  
  
There can be abig failure in case when during the write process the network gets failed or all the nodes have crashed.because if during write if one of the node gets crashed then the master will tell to write to some other node.  
  
  
**H2 (2 masters)** Due to failure a new concept of multiple masters has been come.  
There will be one active master na dother passive master nodes.

There will be zookeeper who will keep a check on all the masters.If an active master is down it will make the othe rpasive master as the active master.  
  
  
Suppose we know that the active master node will create metadata then what happens if the active node goes down what happens to the metadata.  
Now there is a concept of zonal node where all the metadata will be kept.

**SPACE QUOTA**

This is much more towards the admin side where ona node in hdfs they set how much data you can upload.  
  
**NORMAL QUOTA**

It will tell how many files allowed ina directory

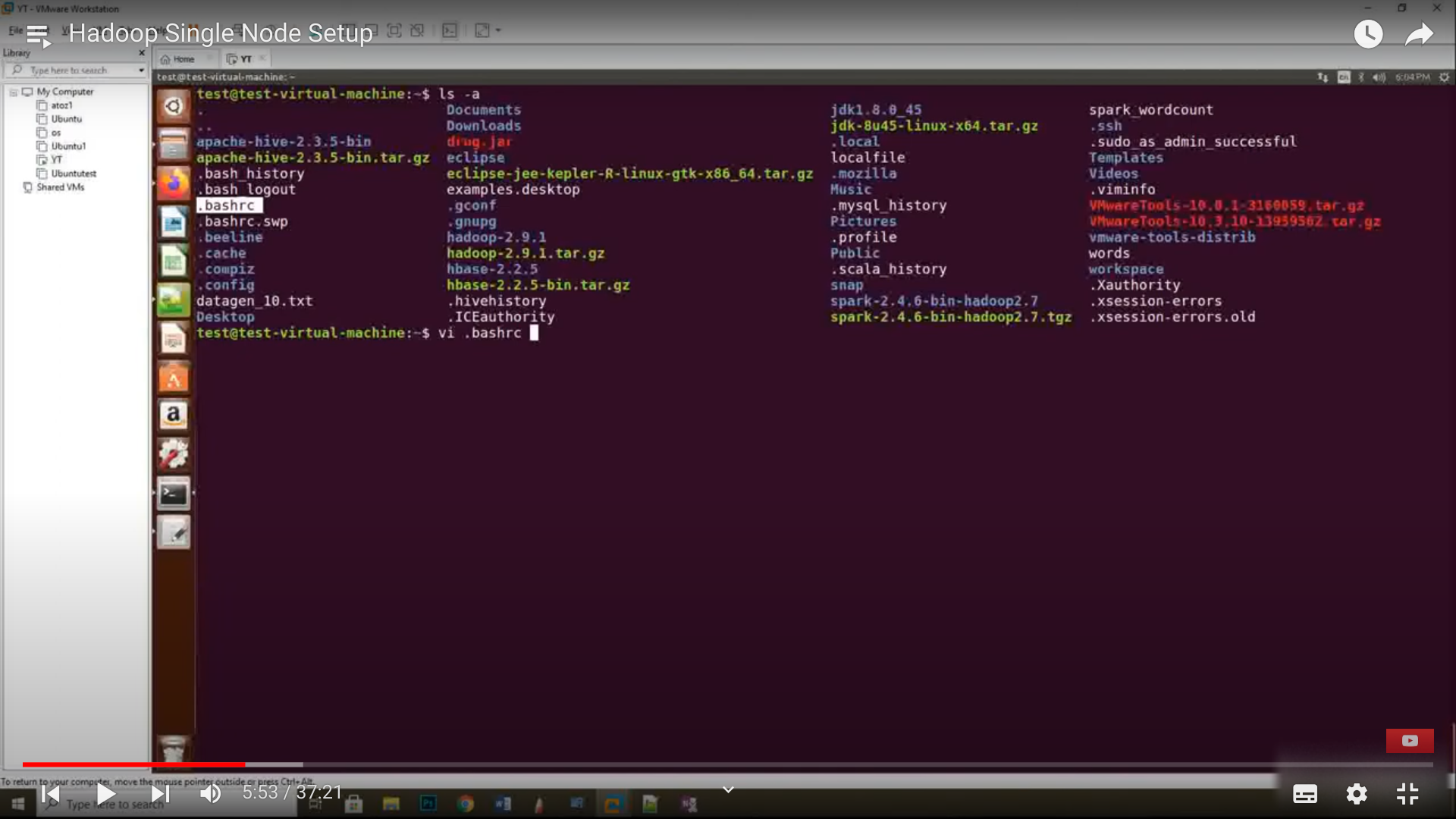
**HADOOP SETUP**

1. **First install**  from the website

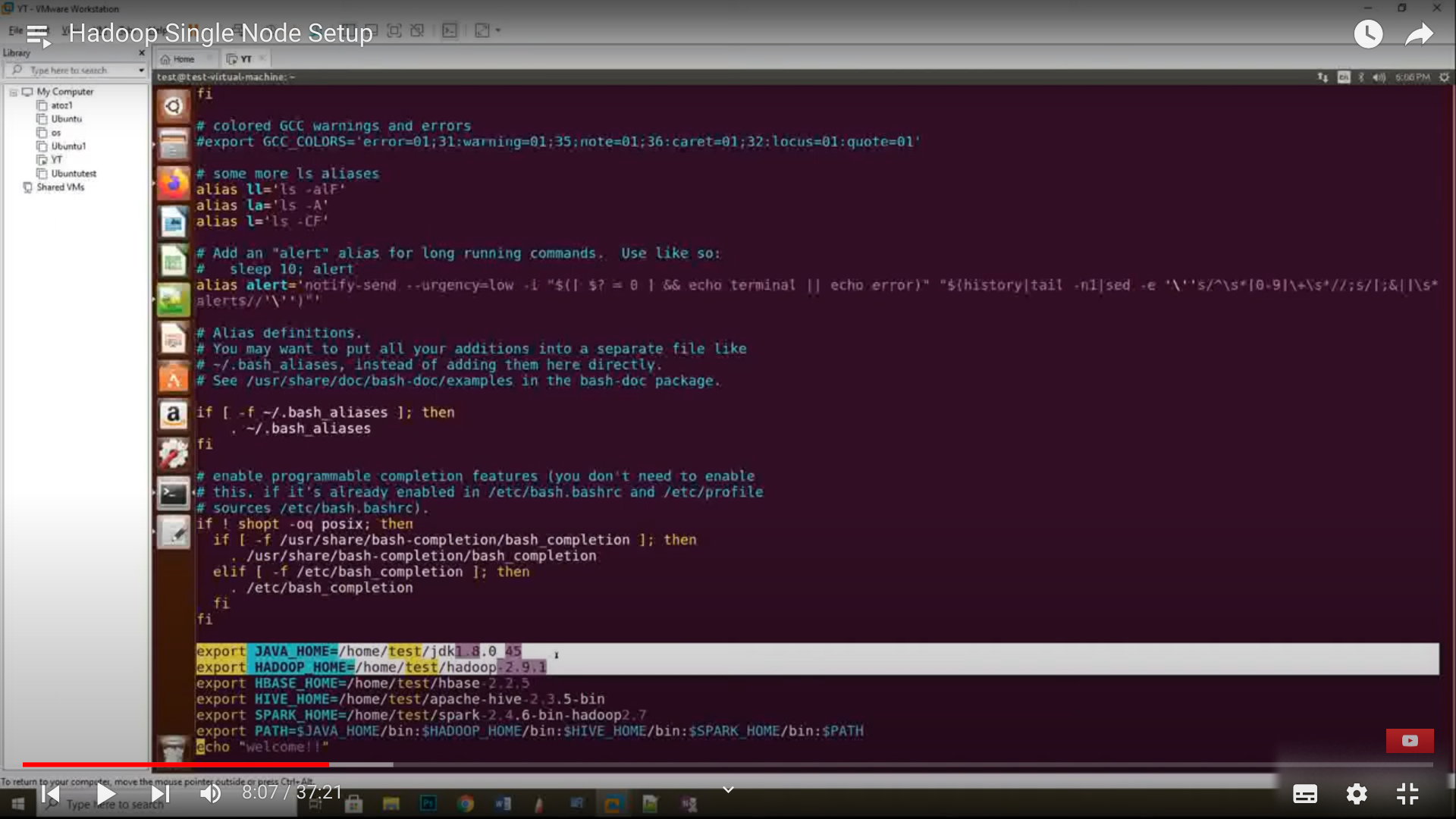


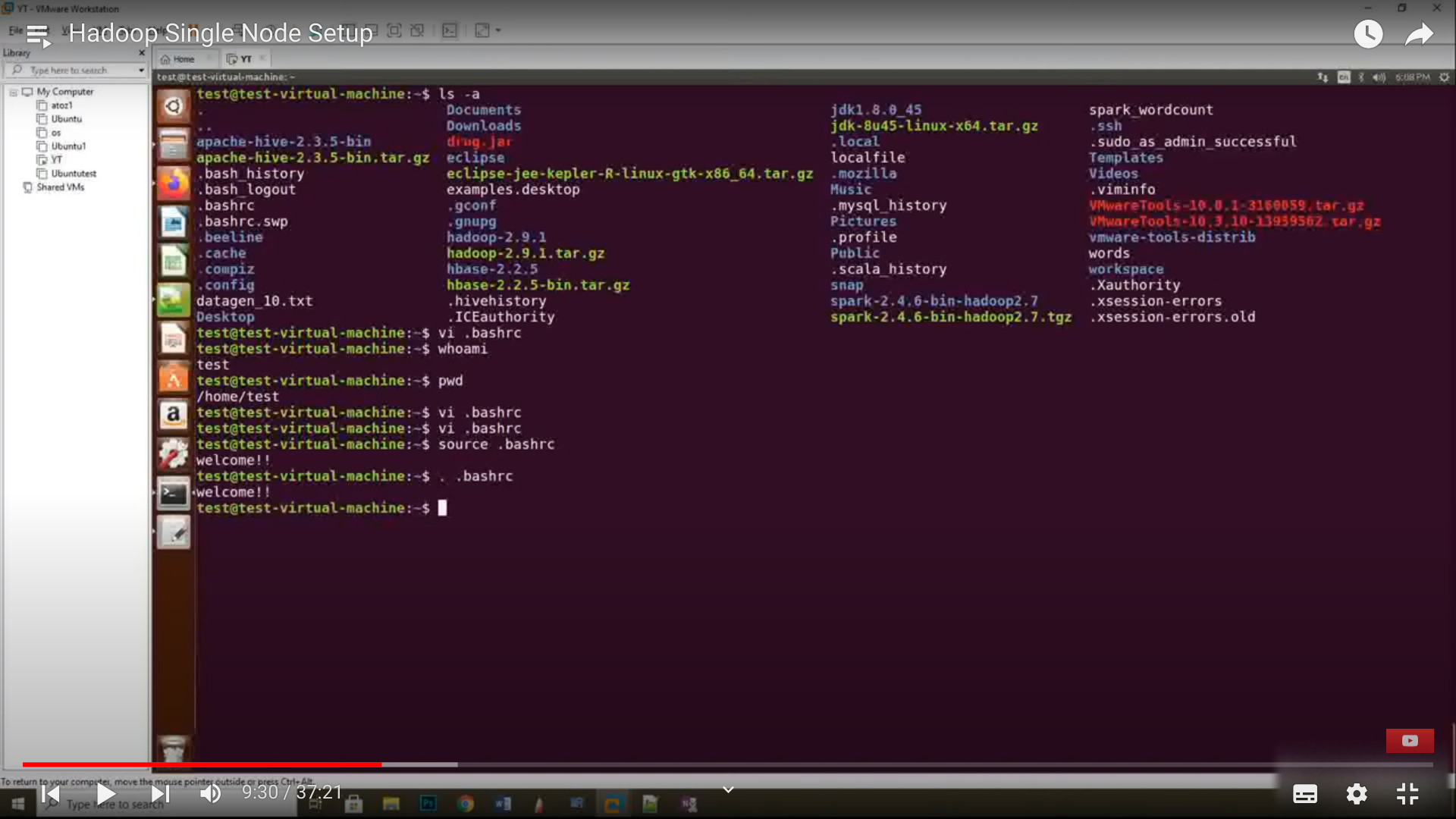
2.download jdk 1.8

3.do vi .bashrc to do environmental setup

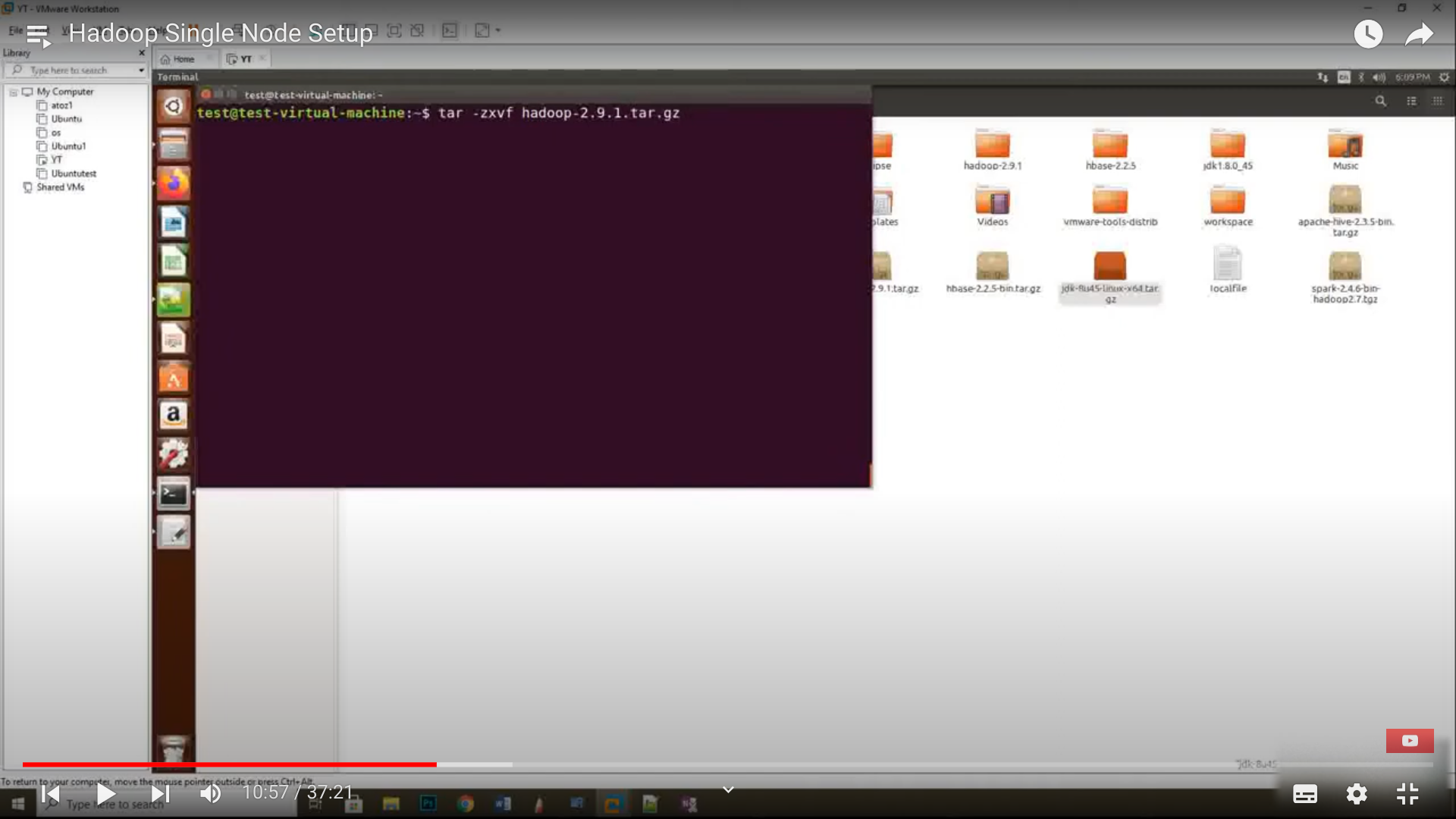
****

**4.inside this file give your java home an dhadoop home**

****

**5.** After making the required changes simply do 

6.java setup is doen now will do the hadoop setup



**Complete steps**

**Youtube.com/atozknowledgevideos**

**Hadoop-2 Single Node Cluster Creation Installation**

**-----------------------------------------------------------------**

**1. Download Hadoop and Java**

**tar -zxvf hadoop-2.9.1.tar.gz (Extract the tar file)**

**tar -zxvf jdk-8u45-linux-x64.tar (Extract the tar file)**

**sudo apt-get install vim (Install USER Friendly Editer)**

**vi .bashrc (Set the java Path in your Home Path)**

**export JAVA\_HOME=/home/username/jdk1.8.0\_45**

**export PATH=HOME/bin:JAVA\_HOME/bin:PATH**

**source .bashrc (Execute the bashrc file)**

**echo JAVA\_HOME (Check the java path)**

**=========================================================================================================================================**

**2. Modify Hadoop Configuration Files**

**NAMENODE ----> core-site.xml**

**RESOURCE MANGER ----> mapperd-site.xml**

**SECONDARYNAMENODE ---->**

**DATANODE ----> slaves**

**NODEMANGER ----> slaves & yarn-site.xml**

**vi etc/hadoop/core-site.xml**

**<property>**

**<name>fs.default.name</name>**

**<value>hdfs://localhost:50000</value>**

**</property>**

**vi etc/hadoop/yarn-site.xml**

**<property>**

**<name>yarn.nodemanager.aux-services</name> <value>mapreduce\_shuffle</value>**

**</property>**

**<property>**

**<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name> <value>org.apache.hadoop.mapred.ShuffleHandler</value>**

**</property>**

**<property>**

**<description>The hostname of the RM.</description>**

**<name>yarn.resourcemanager.hostname</name>**

**<value>localhost</value>**

**</property>**

**<property>**

**<description>The address of the applications manager interface in the RM.</description>**

**<name>yarn.resourcemanager.address</name>**

**<value>{yarn.resourcemanager.hostname}:8032</value>**

**</property>**

**vi etc/hadoop/hdfs-site.xml**

**<property>**

**<name>dfs.namenode.name.dir</name>**

**<value>/home/username/hadoop2-dir/namenode-dir</value>**

**</property>**

**<property>**

**<name>dfs.datanode.data.dir</name>**

**<value>/home/username/hadoop2-dir/datanode-dir</value>**

**</property>**

**vi etc/hadoop/mapred-site.xml**

**<property>**

**<name>mapreduce.framework.name</name>**

**<value>yarn</value>**

**</property>**

**vi etc/hadoop/hadoop-env.sh**

**export JAVA\_HOME=/home/username/jdk1.8.0\_45**

**vi etc/hadoop/mapred-env.sh**

**export JAVA\_HOME=/home/username/jdk1.8.0\_ 45**

**vi etc/hadoop/yarn-env.sh**

**export JAVA\_HOME=/home/username/jdk1.8.0\_45**

**vi etc/hadoop/slaves**

**localhost**

**Install the ssh key**

**(Generates, Manages and Converts Authentication keys)**

**sudo apt-get install openssh-server**

**ssh-keygen -t rsa**

**(Setup passwordless ssh to localhost and to slaves )**

**cd .ssh**

**ls**

**cat id\_rsa.pub >> authorized\_keys (copy the .pub)**

**(Copy the id\_rsa.pub from NameNode to authorized\_keys in all machines)**

**ssh localhost**

**(Asking No Password )**

**=========================================================================================================================================**

**3. Format NameNode**

**cd hadoop-2.9.1**

**bin/hadoop namenode -format (Your Hadoop File System Ready)**

**=========================================================================================================================================**

**4. Start All Hadoop Related Services**

**sbin/start-all.sh**

**(Starting Daemon’s For DFS & YARN)**

**NameNode**

**DataNode**

**SecondaryNameNode**

**ResourceManager**

**NodeManager**

**(check the Browser Web GUI )**

**NameNode - http://localhost:50070/**

**Resource Manager - http://localhost:8088/**

**=========================================================================================================================================**

**5.Stop All Hadoop and Yarn Related Services**

**sbin/stop-all.sh**