Running Kmeans Mapreduce code on Amazon AWS

Pseudo Code

Input: Dataset *D*, Number of clusters *k* **Output:** Data points with cluster memberships

Step 1: for *iteration* = 1 to *MaxIterations* do

- **Step 2**: Mapper: Read *D* and centroids from HDFS
- **Step 3**: Mapper: Compute the distance between each point in D and each point in *centroids*
- Step 4: Mapper Output: Key-value pairs with key as centroid id and value as data point id and distance between them
- **Step 5**: Shuffle and Sort: Aggregate for each key (centroid)
- Step 6: Reducer: Sort distances and associate data points to the nearest centroid
- Step 7: Reducer: Recompute the *centroids*
- Step 8: Reducer Output: Write centroids to HDFS

Step 9: end for

How to run the code

Tools Required

- 1. Amazon AWS Account
- 2. <u>PuTTy</u> Windows Client (to connect to Amazon EC2 instance)
- 3. <u>PuTTYgen</u> (to generate private key this will be used in putty to connect to EC2 instance)
- 4. <u>WinSCP</u> (secury copy)

1. Setting up Amazon EC2 Instances

With 4 node clusters and minimum volume size of 8GB there would be an average \$2 of charge per day with all 4 running instances. You can stop the instance anytime to avoid the charge, but you will loose the public IP and host and restarting the instance will create new ones. You can also terminate your Amazon EC2 instance anytime and by default it will delete your instance upon termination, so just be careful what you are doing.

1.1 Get Amazon AWS Account

If you do not already have an account, please create a new one. Amazon EC2 comes with eligible free-tier instances.

1.2 Sign In to AWS

Once login into Amazon Web Services website (<u>http://aws.amazon.com</u>) by clicking *Sign In to the Console*

E Menu	webservices	English Very My Account Sign In to the Console
	Explore the new services and features announced at the New York Summit See what's new *	
	Sign in to t	he Console
	Sign In to th	e Console

After successful signing in to AWS, click EC2



Then click Launch Instance



1.3 Select AMI

Under the *Quick Start* column on the left, click *Community AMIs*. search and select Ubuntu Server 12.04 Server 64-bit OS

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start		0 uburbu 40 0	< < 1 to 50 of	6,073 AMIs 🖒 🔀
My AMIs			i X	
AWS Marketplace		wawsmar 🐫	ketplace 121 results for "ubuntu 12.04" on AWS Marketplace Partner software pre-configured to run on AWS	
Community AMIs		0	ubuntu/images/ebs/ubuntu-precise-12.04-amd64-server-	Select
Operating system		e	Root device type: ebs Virtualization type: paravirtual	64-bit
Cent OS	* ©	0	ubuntu/images/ebs/ubuntu-precise-12.04-amd64-server- 20140923.1 - ami-00ab0168	Select
 Fedora Gentoo 	6 9		Root device type: ebs Virtualization type: paravirtual	64-bit

1.4 Select Instance Type

Select the micro instance and click on 'Next: Configure Instance Details'; on bottom right.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run a resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter	ilter by: All instances 👻 Current generation 👻 Show/Hide Columns										
Currently selected: t1.micro (up to 2 ECUs, 1 vCPUs, 0.613 GiB memory, EBS only)											
	Family		Туре -	ECUs (i) -	vCPUs (j) -						
	Micro instances Free tier eligible		t1.micro	up to 2	1						

1.5 Configure Number of Instances

We are setting up 4 node Hadoop cluster, so please enter 4 as number of instances. Please check Amazon EC2 free-tier requirements, you may setup 3 node cluster with < 30GB storage size to avoid any charges. In production environment you want to have SecondayNameNode as separate machine. Click on 'Next: Add Storage'; at bottom right.

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take

Number of instances	(i)	4
Purchasing option	(i)	Request Spot Instances
Network	(i)	vpc-f839fb9d (172.31.0.0/16) (default)
Subnet	()	No preference (default subnet in any Availability Zor Create new subnet
Public IP	()	Automatically assign a public IP address to your instances
IAM role	i	None
Shutdown behavior	i	Stop v
Enable termination protection	()	Protect against accidental termination
Monitoring	()	Enable CloudWatch detailed monitoring
		Additional charges apply.
Tenancy	()	Shared tenancy (multi-tenant hardware)
		Additional charges will apply for dedicated tenancy.

1.6 Add Storage

Minimum volume size is 8GB. Change it to 20GB (since we can add upto 30GB in free tier) and also change the volume type to "General Purpose (SSD)". Click on 'Next: Tag Instance'; at bottom right.

Step 4: Add S	torage						
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.							
Туре ()	Device (i)	Snapshot (j)	Size (GiB) (j	Volume Type (j)	IOPS (j	Delete on Termination (i)	Encrypted
Root	/dev/sda1	snap-49967039	20	General Purpose (SSD) V	60 / 3000		Not Encrypted

1.7 Instance Description

Give your instance name as "HadoopEC2MultiNodeCluster" and click on 'Next: Configure Security group'; at bottom right.

Step 5: Tag Instance



1.8 Define a Security Group

Create a new security group, later on we are going to modify the security group and add new rules which will allow seamless interaction between instances. Name it 'HadoopEC2SecurityGroup'. Click *Review and Launch* at bottom right.

Step 6: Configure Security Group

A security group is a set of firewall rules that co the HTTP and HTTPS ports. You can create a r	ntrol the traffic for your instance. On this page, you can add rules to allow specific i new security group or select from an existing one below. Learn more about Amazor	raffic to reach your instance. For example, if you want EC2 security groups.
Assign a security group:	Create a new security group	
	Select an existing security group	
Security group name:	HadoopEC2SecurityGroup	
Description:	launch-wizard-14 created on Wednesday, June 18, 2014 1:19:57 PM UTC-4	
Туре ()	Protocol (j)	Port Range (i)
	TOD	00
35H ·	ICP	22
Add Rule	ICP	22

1.9 Launch Instance and Create Security Pair

Review and Launch Instance.

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.

Create a new keypair and give it a name "hadoopec2cluster" and download the keypair (.pem) file to your local machine. Click Launch Instance

Select	an existing key pair or create a new key pair	>						
A key pair they allow to obtain th securely S	consists of a public key that AWS stores, and a private key file that you to connect to your instance securely. For Windows AMIs, the priva he password used to log into your instance. For Linux AMIs, the private SH into your instance.	you store. Together, te key file is required key file allows you to						
Create	e a new key pair	¥						
Key pa	air name							
Select an existing key pair or create a new key pair A key pair consists of a public key that AWS stores, and a private key file that you store. To they allow you to connect to your instance securely. For Windows AMIs, the private key file is re- to obtain the password used to log into your instance. For Linux AMIs, the private key file allows securely SSH into your instance. Create a new key pair Key pair name hadoopec2cluster You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created. Cancel Launch Instan								
	Dow	nload Key Pair						
	You have to download the private key file (*.pem file) before you ca Store it in a secure and accessible location. You will not be able file again after it's created.	an continue. to download the						
	Cancel	aunch Instances						

1.10 Launching Instances

Once you click "Launch Instance" 4 instance should be launched with "pending" state

Lau	nch Instance Con	nect	Actions 👻					
Filte	r: All instances 🜱	All instar	nce types 👻	Q Search Instance	es	×		K < 1 to 4 c
	Name	9.	Instance ID ·	Instance Type ~	Availability Zone -	Instance State ~	Status Checks 👻	Alarm Status ~
\odot	HadoopEC2Cluster		i-a8d5b388	t1.micro	us-east-1b	🥚 pending	🗶 Initializing	None 🍃
\odot	HadoopEC2Cluster		i-a9d5b389	t1.micro	us-east-1b	pending	🛛 Initializing	None 🍃
\odot	HadoopEC2Cluster		i-aad5b38a	t1.micro	us-east-1b	🥚 pending	📓 Initializing	None 🍗
\odot	HadoopEC2Cluster		i-abd5b38b	t1.micro	us-east-1b	pending	🛛 Initializing	None 🍃

Once in "running" state we are now going to rename the instance name as below.

- 1. HadoopNameNode (Master)
- 2. HadoopSecondaryNameNode
- 3. HadoopSlave1 (data node will reside here)
- 4. HaddopSlave2 (data node will reside here)

You can rename the instance by clicking by hovering on the name and clicking on the pen icon showed next to it. Once renamed click on the tick mark.

Filter	: All instances 👻 All instar	nce types 👻	Q, Search Instance	es	×		< < 1 t	o 4 o	f
	Name 🛛 🖓 -	Instance ID +	Instance Type 👻	Availability Zone -	Instance State ~	Status Checks 👻	Alarm Status	÷	-
	HadoopNameNonde	i-a8d5b388	t1.micro	us-east-1b	running	🤣 2/2 check	None	7	e
	HadoopSecondayNameNode	i-a9d5b389	t1.micro	us-east-1b	running	🤣 2/2 check	None	6	ę
	HadoopSlave1	i-aad5b38a	t1.micro	us-east-1b	running	🥝 2/2 check	None	20	6
	HadoopSlave2	i-abd5b38b	t1.micro	us-east-1b	running	🥝 2/2 check	None	7	6

Please note down the Instance ID, Public DNS/URL like (ec2-54-209-221-112.compute-1.amazonaws.com) and Public IP for each instance for your reference. We will need it later on to connect from Putty client. Also notice we are using "HadoopEC2SecurityGroup".

Filte	r: All instances 👻 All instar	nce types 👻	Q Search Instance	25	×		< < 1 to
	Name 🌳 🗸	Instance ID *	Instance Type 👻	Availability Zone -	Instance State ~	Status Checks 👻	Alarm Status
	HadoopNameNonde	i-a8d5b388	t1.micro	us-east-1b	running	2/2 check	None 🏷
	HadoopSecondayNameNode	i-a9d5b389	t1.micro	us-east-1b	running	2/2 check	None 🍾
	HadoopSlave1	i-aad5b38a	t1.micro	us-east-1b	running	2/2 check	None 🍾
	HadoopSlave2	i-abd5b38b	t1.micro	us-east-1b	🔵 running	2/2 check	None 🏷

Instance: i-a8d5b388 (HadoopNameNonde) Public DNS: ec2-54-209-221-112.compute-1.amazonaws.com

Description	Status Checks	Monitoring	Tags		
	Instance ID	i-a8d5b388		Public DNS	ec2-54-209-221-112.compute- 1.amazonaws.com
	Instance state	running		Public IP	54.209.221.112
	Instance type	t1.micro		Elastic IP	
	Private DNS	ip-172-31-35-98.e	c2.internal	Availability zone	us-east-1b
	Private IPs	172.31.35.98		Security groups	HadoopEC2SecurityGroup. view
Second	dary private IPs			Scheduled events	No scheduled events
	VPC ID	vpc-120e1470		AMI ID	ubuntu-precise-12.04-amd64-se (ami-a73264ce)

Modify Security Group

You can use the existing group or create a new one. When you create a group with default options it add a rule for SSH at port 22. In order to have TCP and ICMP access we need to add 2 additional security rules. Add 'All TCP', 'All ICMP' and 'SSH (22)' under the inbound rules to "HadoopEC2SecurityGroup". This will allow ping, SSH, and other similar commands among servers and from any other machine on internet. Make sure to "Apply Rule changes" to save your changes.

These protocols and ports are also required to enable communication among cluster servers. As this is a test setup we are allowing access to all for TCP, ICMP and SSH and not bothering about the details of individual server port and security.

Please note that without adding these rules, the instances in Hadoop cluster with default security group won't be able to communicate between each other's.

Crea	ate Securi	ty Group Delete				୯	٥	6
Viewin	g: All Se	curity Groups • Search)			 1 to 2 of 2 Ite	ms	>
	Group ID	Name	VPC ID	Description				
	sg-0436c8	61 🄌 default	vpc-120e1470	default VPC security grou	IP			
	sg-0811ef6	id HadoopEC2SecurityGroup	vpc-120e1470	security group for hadoop	ec2 security group			
1 Sec Det	surity Grou Security	or selected Group: HadoopEC2Secur bound* Outbound	ityGroup			8 8	3	
Cre	eate a	Custom TCP rule		CMP rt (Service)	Saurca	Actio		
Po	rt range:		ALI		0.0.0/0	Delete		
	re range.	(e.g., 80 or 49152-65535)	Т	СР				
So	urce:	0.0.0.0/0	Po	rt (Service)	Source	Action	1	
		(e.g., 192.168.2.0/24, sg-47ad482e,	or 22	(SSH)	0.0.0/0	Delete		
		1234567690/delauit)	0 -	65535	0.0.0/0	Delete		
	Your o	hanges have not been applied y	vet.					
		Apply Rule Changes						

2. Setting up client access to Amazon Instances

Now, let's make sure we can connect to all 4 instances. For that we are going to use Putty client. We are going setup password-less SSH access among servers to setup the cluster. This allows remote access from Master Server to Slave Servers so Master Server can remotely start the Data Node and Task Tracker services on Slave servers.

We are going to use downloaded hadoopec2cluster.pem file to generate the private key (.ppk). In order to generate the private key we need Puttygen client. You can download the putty and puttygen and various utilities in zip from <u>here</u>.

2.1 Generating Private Key

Let's launch PUTTYGEN client and import the key pair we created during launch instance step – "hadoopec2cluster.pem"

Navigate to Conversions and "Import Key"

File Key Conversio	ons) Help	0								
Key Imp	ort key									
No key. Expo	ort OpenS	SH key								
Expo	ort ssh.cor	m key								
Actions Generate a public/pri Load an existing priva Save the generated k Parameters Type of key to genera SSH-1 (RSA) Number of bits in a ge	vate key pa ate key file tey ate: @ enerated ke	air Sa SSH-2 RSA ay:	ve public key	Generate Load Save private key SSH-2 DSA 2048						
File Key Convers	ator ions He	elp		8	23				_	×
File Key Convers Load private key:	ator ions He ws EC2	elp	-	ę	23		Search a	ws EC2		×
File Key Convers Load private key:	ator ions He ws EC2 ew folder	elp		8	23	▼ € ₂	Search a	ws EC2 ₿☷ ▼		× , (2)
File Key Convers Load private key:	ator iions He ws EC2 ew folder	elp Name	-	9 Date	22	▼ €ŋ Type	Search a	ws EC2 ₿☷ ▼		×
File Key Convers Load private key: Convers Corganize Key Corganize Key	ator ions He ws EC2 ew folder	elp Name	cluster	ହ Date 1/10/2014 6:04 PM	23	▼ 4y Type PEM File	Search a	ws EC2		× , @
File Key Convers Load private key: Coad private key: Corganize ▼ N Favorites Desktop Downloads	ator ions He ws EC2 ew folder	elp Name I hadoopec2	cluster	2 Date 1/10/2014 6:04 PM	22	← ← Type PEM File	Search a	ws EC2 ≌≕ ▼		×
Pull IY Key Generative File Key Converse Load private key: Image: Converse Imag	ator ions He ws EC2 ew folder s	elp Name Adoopec2	cluster	9 Date 1/10/2014 6:04 PM	22	Type PEM File	Search a	ws EC2		× , @
Purity Key Generative File Key Convers Load private key: Organize ▼ N Favorites Desktop Downloads Recent Place	ator ions He ws EC2 ew folder s	elp Name Andoopec2	cluster	₽ Date 1/10/2014 6:04 PM	23	▼ 49 Type PEM File	Search a	ws EC2		× ,, @
Purity Key Generative File Key Convers Coad private key: Coad private key: Corganize ▼ N Favorites Desktop Downloads Recent Place Comparise Documents	ator ions He ws EC2 ew folder s E	elp Name	cluster	2 Date 1/10/2014 6:04 PM	23	← 49 Type PEM File	Search a	ws EC2		× ×
File Key Convers Load private key: Convers Load private key: Convers Corganize N Corganize	ator ions He ws EC2 ew folder s	elp Name I hadoopec2	cluster	2 Date 1/10/2014 6:04 PM	22	▼ €y Type PEM File	Search a	ws EC2	iew avail	×
Purity Key Generative File Key Convers Load private key: Corganize ▼ N Favorites Desktop Downloads Recent Places Documents Music Pictures	ator ions He ws EC2 ew folder s E	Name	cluster	P Date 1/10/2014 6:04 PM	23	▼ 4y Type PEM File	Search a	ws EC2	iew avail	X J Ø
Pull IY Key Generative File Key Load private key: Image: Conversion of the second private key: Image: Conversion of the secon	ator ions He ws EC2 ew folder s E	elp Name	cluster	P Date 1/10/2014 6:04 PM	23	▼ 49 Type PEM File	Search a	ws EC2	iew avail	X J Q
Purity Key Genera File Key Convers Load private key: Corganize ▼ N Favorites Desktop Downloads Recent Places Libraries Documents Music Pictures Videos Homegroup	ator ions He ws EC2 ew folder s E	elp Name	cluster	P Date 1/10/2014 6:04 PM	23	▼ fy Type PEM File	Search a	ws EC2	iew avail	× ۶
Pull IY Key Generative File Key Converse Load private key: Image: Converse Image: Converse Image: Computer Image: Computer	ator ions He ws EC2 ew folder s	elp Name I hadoopec2	cluster	P Date 1/10/2014 6:04 PM		▼ 49 Type PEM File	Search a	ws EC2	iew avail	X P Q
Pull Y key Genera File Key Convers Load private key: Corganize ▼ N Favorites Desktop Downloads Recent Places Documents Music Pictures Videos Homegroup Computer	ator ions He ws EC2 ew folder s E	elp Name I hadoopec2	cluster	P Date 1/10/2014 6:04 PM		▼ fy Type PEM File	Search a	ws EC2	iew avail	× ¢
Fle Key Convers Load private key: Load private key: Convers Conganize Favorites Desktop Downloads Recent Place Libraries Documents Music Pictures Videos Homegroup Computer	ettor ions He ws EC2 ew folder s E File nar	elp Name Adoopec2	cluster	P Date 1/10/2014 6:04 PM		▼ fy Type PEM File	Search a	ws EC2	iew avail	× e able.

Once you import the key you can enter passphrase to protect your private key or leave the passphrase fields blank to use the private key without any passphrase. But for now leave it **blank**. Passphrase protects the private key from any unauthorized access to servers using your machine and your private key.

Any access to server using passphrase protected private key will require the user to enter the passphrase to enable the private key enabled access to AWS EC2 server.

2.2 Save Private Key

Now save the private key by clicking on "Save Private Key" and click "Yes" as we are going

to leave passphrase empty.

•	DTTV //	Concentra	
ile Key Conversion	P	uTTYgen Warnin	g 🛛 🗙
Key <u>P</u> ublic key for pasting in ssh-rsa AAAAB3NzaC1yc2EAA	Are yo withou	u sure you want to sa it a passphrase to pro	ave this key otect it?
OMaKNEArMijsal93HM qul//9zQ7d5gQRvrOb +3BCLd7+NiWtbwWzrl		Yes	No
Key fingerprint:			
Key comment:	mported-openssh-key		
Key p <u>a</u> ssphrase:			
Confirm passphrase:			
Actions			
Generate a public/privat	e key pair		<u>G</u> enerate
Load an existing private	key file		<u>L</u> oad
Save the generated key		Save p <u>u</u> blic key	<u>S</u> ave private key
Parameters			
Type of key to generate O SSH- <u>1</u> (RSA)	● SSH-2 <u>R</u> SA	⊖ ss⊦	1-2 <u>D</u> SA

Save the .ppk file and give it the same name.

😴 PuTTY Key Generat	tor	8 2	3		ncategorized
File Key Conversio	ons Help			+ Add N	ew Category
Save private key as:	- Institution and				X
OO~ I + aw	rs EC2		✓ Sec	arch aws EC2	Q
Organize 👻 New	w folder			: :=:	• 📀
☆ Favorites	▲ Name	Date	Туре	Size	Tags
 Desktop Downloads Recent Places Libraries Documents Music Pictures Videos 	E	No items match	your search.		
輚 Homegroup	-	III			•
File <u>n</u> ame:	haddopec2cluster				•
Save as <u>t</u> ype:	PuTTY Private Key Files (*.ppk)				•
) Hide Folders				<u>S</u> ave	Cancel

Now we are ready to connect to our Amazon Instance Machine for the first time.

2.3 Connect to Amazon Instance

Let's connect to HadoopNameNode first. Launch Putty client, grab the public URL (the DNS ec2-...-amazonaws.com from the console step 1.10), import the .ppk private key that we just created for password-less SSH access. As per amazon documentation, for Ubuntu machines username is "ubuntu"

2.3.1 Provide private key for authentication



2.3.2 Hostname and Port and Connection Type

Host name will be like "Ubuntu@ec2-.....compute-1.amazonaws.com"

Click "Open" to launch putty session.

8	PuTTY Configuration	×
Category: 	Basic options for your PuTTY session Specify the destination you want to connect to Host Name (or IP address) Port uburtu@ec2-54-88-22-174.compute-1.ar 22 Connection type: Raw Ielnet Rlogin Sst Setjal Load, save or delete a stored session Savgd Sessions	
About	<u>Open</u> <u>Cancel</u>	

When you launch the session first time, you will see below message, click "Yes" 23 - 0 Rec2-54-209-221-112.compute-1.amazonaws.com - PuTTY X **PuTTY Security Alert** The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is. The server's rsa2 key fingerprint is: ssh-rsa 2048 dd:29:2d:75:81:83:22:d1:49:b9:bc:ce:fc:e9:89:22 If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting. If you want to carry on connecting just once, without adding the key to the cache, hit No. If you do not trust this host, hit Cancel to abandon the connection. Yes No Cancel Help

If everything goes well you will be presented welcome message with Unix shell at the end.



If there is a problem with your key, you may receive below error message. Check manual again and eradicate the error.



Similarly connect to remaining 3 machines HadoopSecondaryNameNode, HaddopSlave1 and HadoopSlave2 respectively to make sure you can connect successfully.



2.4 Enable Public Access

Issue **ifconfig** command in the terminal and note down the ip address. Next, we are going to update the hostname with ec2 public URL and finally we are going to update /etc/hosts file to map the ec2 public URL with ip address. This will help us to configure master and slaves nodes with hostname instead of ip address.

Following is the output on HadoopNameNode ifconfig



This IP is same as what we have in the console

А	В	E
AMI name	public dns	inet address
HadoopNameNode	ec2-54-209-221-112.compute-1.amazonaws.com	172.31.35.98

Now, issue the hostname command, it will display the ip address same as inet address from ifconfig command.



We need to modify the hostname to ec2 public URL with below command

~\$ sudo hostname ec2......compute-1.amazonaws.com (Please put the URL which you got)



2.5 Modify /etc/hosts

Let's change the host to EC2 public IP and hostname.

Open the /etc/hosts in 'VI' with "sudo vi /etc/hosts" command, in a very first line it will show 127.0.0.1 localhost, we need to replace that with amazon ec2 hostname and ip address we just collected.



Modify the file and save your changes. To save a file you need to type the following sequence

- 1. Esc, to quit edit mode and fallback to command mode
- 2. wq, start with a colon and then press w and q to write and quit
- 3. Then press Enter to validate.



Repeat 2.3 and 2.4 sections for remaining 3 machines.

3. Setup WinSCP access to EC2 instances

In order to securely transfer files from your windows machine to Amazon EC2 <u>WinSCP</u> is a handy utility.

Provide hostname, username and private key file and save your configuration and Login (Same credentials which were used for putty login)

WinSCP Login	
New Site Session Directories Authentication entrely (Sin-2)	
HadoopMaster-ubuntu@ec2-54-203 Ele protocol:	
Hadooptianeholde	
HadopSlave2-ubuntu@ec2-5+244 Host name: Port number:	
HadoopS/N-kauntu@ec2-54-203-2: 12-54-209-221-112.compute-1.amazonaws.com 22	
SSH SSH	
2. Provide Username ubuntu	
Bugs Allow agent forwarding	
Save Cancel Advanced	r nak"
3 Provide ppk file	-bibut
GSSAPI	
Attempt GSSAPI authentication (SSH-2)	
Allow GSSAPI credential delegation	
< <u> </u>	
Tools V Manage V Close Help	
Color OK Cancel	Help
Warning	
File "C haddones?cluster.nnk"	
does not evit or it does not contain private levin la moduperactuater.ppk	
does not exist on t does not contain private key in known romat.	
Abort Ignore Help	

If you see above error, just ignore and you upon successful login you will see unix file system of a logged in user /home/ubuntu your Amazon EC2 Ubuntu machine.

Sa Windows - HadoopNam	eNode - WinSCP	
Local Mark Files Com	mands Session Opt	ions Remote Help
🖶 🎛 📚 Synchronize	🗩 🦑 👔	🔛 🗃 Queue 👻 Transfer Settings 🛛 👋 🥵 🗸
📮 HadoopNameNode 🛛	🖆 New Session	
🏭 🖓 🕶 🚰 🔽 🖛 🗸	⇒ - 🔁 🔽 »	📔 ut 🝷 😰 (+ + +) + 💼 🖬 🏠 🥭 👋
🔋 📑 Upload 📑 📝 Edit	🗙 🛃 » 主 »	🛙 🔐 Download 🔐 📝 Edit 🗙 🚮 🕞 Properties 👋 🛛 🛨 🂙
C:\Windows		/home/ubuntu
Name Êxt	Size Ty ^	Name Ext
1	Pa	🕹
\$CrmUninstallKB2739	Fil	🕌 .cache
\$CrmUninstallKB2739	Fil	📕 .ssh
\$CrmUninstallKB2791	Fil	bash_history
CrmUninstallKB2791	Fil	viminfo
SCrmUninstallKB2795	Fil	.sudo_as_admin_successful
SCrmUninstallKB2795	Fil	🖬 .profile
SCrmUninstallKB2843	Fil	bashrc
🔋 🛯 \$CrmUninstallKB2843	Fil	bash_logout
SCrmUninstallKB2849	Fil 🖕	
III	•	۰
0 B of 19,864 KiB in 0 of 154		0 B of 5,252 B in 0 of 8
		G SFTP-3 🧐 0:00:50

Upload the .pem file to master machine (HadoopNameNode). It will be used while connecting to slave nodes during hadoop startup daemons.

1. Apache Hadoop Installation and Cluster Setup

1.1 Update the packages and dependencies.

Let's update the packages, I will start with master, **repeat this for SecondaryNameNode and 2 slaves**.

Open the connection to the MatserNode using the steps provided in '2.3 Connect to Amazon Instance'

Type the following in the opened terminal

\$ sudo apt-get update

Once it's complete, let's install java

1.2 Install Java

Add following PPA and install the latest Oracle Java (JDK) 7 in Ubuntu

\$ sudo add-apt-repository ppa:webupd8team/java

Then type

\$ sudo apt-get update && sudo apt-get install oracle-jdk7-installer

Check if Ubuntu uses JDK 7

Type:

\$ java -version

The response should be like this



Repeat this for SNN and 2 slaves.

1.3 Download Hadoop

I am going to use haddop 1.2.1 stable version from apache download page and here is the 1.2.1 mirror

Issue wget command from shell

\$ wget http://apache.mirror.gtcomm.net/hadoop/common/hadoop-1.2.1/hadoop-1.2.1.tar.gz



Unzip the files and review the package content and configuration files.

\$ tar -xzvf hadoop-1.2.1.tar.gz

ubuntu@ec2-54-209-221-112:~\$ 1s hadoop-1.2.1 hadoop-1.2.1.tar.gz hadoopec2cluster.pem ubuntu@ec2-54-209-221-112:~\$

For simplicity, rename the 'hadoop-1.2.1' directory to 'hadoop' for ease of operation and maintenance.

\$ mv hadoop-1.2.1 hadoop



1.4 Setup Environment Variable

Setup Environment Variable for 'ubuntu' user

Update the .bashrc file to add important Hadoop paths and directories.

Navigate to home directory

\$ cd

Open .bashrc file in vi edit

\$ vi .bashrc

Add following at the end of file

export HADOOP_CONF=/home/ubuntu/hadoop/conf export HADOOP_PREFIX=/home/ubuntu/hadoop

#Set JAVA_HOME

export JAVA_HOME=/usr/lib/jvm/java-7-oracle

Add Hadoop bin/ directory to path export PATH=\$PATH:\$HADOOP_PREFIX/bin

Save and Exit by pressing escape and typing ':wq' and enter to validate.

To check whether it has been updated correctly or not, reload bash profile, use following commands

\$ source ~/.bashrc

By typing the following two commands, there should be some value which should come up

\$ echo \$HADOOP_PREFIX

\$ echo \$HADOOP_CONF

Repeat 1.3 and 1.4 for remaining 3 machines (SNN and 2 slaves).

1.5 Setup Password-less SSH on Servers

Master server remotely starts services on salve nodes, which requires password-less access to Slave Servers. AWS Ubuntu server comes with pre-installed OpenSSh server.

Quick Note:

The public part of the key loaded into the agent must be put on the target system in ~/.ssh/authorized_keys. This has been taken care of by the AWS Server creation process

Now we need to add the AWS EC2 Key Pair identity 'HaddopEc2cluster.pem' to SSH profile. In order to do that we will need to use following ssh utilities

- 'ssh-agent' is a background program that handles passwords for SSH private keys.
- 'ssh-add' command prompts the user for a private key password and adds it to the list maintained by ssh-agent. Once you add a password to ssh-agent, you will not be asked to provide the key when using SSH or SCP to connect to hosts with your public key.
 Amazon EC2 Instance has already taken care of 'authorized_keys' on master server, execute following commands to allow password-less SSH access to slave servers.

First of all we need to protect our keypair files, if the file permissions are too open (see below) you will get an error

To fix this problem, we need to issue following commands

\$ chmod 644 authorized_keys

Quick Tip: If you set the permissions to 'chmod 644', you get a file that can be written by you, but can only be read by the rest of the world. \$ chmod 400 haddoec2cluster.pem

Quick Tip: chmod 400 is a very restrictive setting giving only the file onwer read-only access. No write / execute capabilities for the owner, and no permissions what-so-ever for anyone else.

To use ssh-agent and ssh-add, follow the steps below:

1. At the Unix prompt, enter: eval `ssh-agent`

Note: Make sure you use the backquote (`), located under the tilde (~), rather than the single quote (').

2. Enter the command: 'ssh-add hadoopec2cluster.pem'. Make sure you are in the directory where this .pem file is.

It should work this time.

Keep in mind ssh session will be lost upon shell exit and you have repeat ssh-agent and ssh-add commands.

Remote SSH

Let's verify that we can connect into SNN and slave nodes from master

```
ubuntu@ec2-54-209-221-112:~$ ssh ubuntu@ec2-54-209-219-2.compute-1.amazonaws.com
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.2.0-54-virtual x86_64)
 * Documentation: https://help.ubuntu.com/
  System information as of Sun Jan 12 01:00:03 UTC 2014
  System load: 0.0
                                      Processes:
                                                             61

        System load:
        0.0
        Processes:
        61

        Usage of /:
        20.1% of 7.87GB
        Users logged in:
        1

  Memory usage: 27%
                                      IP address for eth0: 172.31.35.95
  Swap usage: 0%
  Graph this data and manage this system at https://landscape.canonical.com/
  Get cloud support with Ubuntu Advantage Cloud Guest:
   http://www.ubuntu.com/business/services/cloud
  Use Juju to deploy your cloud instances and workloads:
    https://juju.ubuntu.com/#cloud-precise
                                                                                          Last login: Sun Jan 12 00:59:39 2014 from ip-172-31-35-98.ec2.internal
ubuntu@ec2-54-209-219-2:~$
```

\$ ssh ubuntu@<your-amazon-ec2-public URL for SNN or any of your slave nodes>

On successful login the IP address on the shell will change.

Type the following to exit from SNN or other nodes and to come back to the master node.

\$ exit

1.6 Hadoop Cluster Setup

This section will cover the hadoop cluster configuration. We will have to modify

- hadoop-env.sh This file contains some environment variable settings used by Hadoop. You can use these to affect some aspects of Hadoop daemon behavior, such as where log files are stored, the maximum amount of heap used etc. The only variable you should need to change at this point is in this file is JAVA_HOME, which specifies the path to the Java 1.7.x installation used by Hadoop.
- core-site.xml key property fs.default.name for namenode configuration for e.g hdfs://namenode/
- hdfs-site.xml key property dfs.replication by default 3
- **mapred-site.xml** key property mapred.job.tracker for jobtracker configuration for e.g jobtracker:8021

We will first start with master (NameNode) and then copy above xml changes to remaining 3 nodes (SNN and slaves)

Finally, in section 1.6.2 we will have to configure conf/masters and conf/slaves.

- **masters** defines on which machines Hadoop will start secondary NameNodes in our multinode cluster.
- **slaves** defines the lists of hosts, one per line, where the Hadoop slave daemons (datanodes and tasktrackers) will run.

Lets go over one by one. Start with masters (namenode).

Perform the following

hadoop-env.sh

\$ vi \$HADOOP_CONF/hadoop-env.sh and add JAVA_HOME shown below and save changes.



core-site.xml

This file contains configuration settings for Hadoop Core (for e.g I/O) that are common to HDFS and MapReduce Default file system configuration property – fs.default.name goes here it could for e.g hdfs / s3 which will be used by clients.

\$ sudo vi \$HADOOP_CONF/core-site.xml

We are going to add two properties

- fs.default.name will point to NameNode URL and port (usually 8020)
- hadoop.tmp.dir A base for other temporary directories. Its important to note that every node needs hadoop tmp directory. I am going to create a new directory "hdfstmp" as below in all 4 nodes. Ideally you can write a shell script to do this for you, but for now going the manual way.
- Perform the following

Exit from core-site.xml

Then

\$ cd

\$ mkdir hdfstmp

Quick Tip: Some of the important directories are dfs.name.dir, dfs.data.dir in hdfssite.xml. The default value for

the dfs.name.dir is f adoop.tmp.dir}/dfs/data and dfs.data.dir is f adoo p.tmp.dir}/dfs/data. It is critical that you choose your directory location wisely in production environment.

Fill the following in the \$HADOOP_CONF/core-site.xml

<configuration> <property> <name>fs.default.name</name> <value>hdfs://ec2-54-209-221-112.compute-1.amazonaws.com:8020</value> </property> <property> <property> <name>hadoop.tmp.dir</name> <value>/home/ubuntu/hdfstmp</value> </property> </configuration>

Save and Exit

hdfs-site.xml

This file contains the configuration for HDFS daemons, the NameNode, SecondaryNameNode and data nodes.

We are going to add 2 properties

- dfs.permissions.enabled with value *false*, This means that any user, not just the "hdfs" user, can do anything they want to HDFS so do not do this in production unless you have a very good reason. if "true", enable permission checking in HDFS. If "false", permission checking is turned off, but all other behavior is unchanged. Switching from one parameter value to the other does not change the mode, owner or group of files or directories. Be very careful before you set this
- **dfs.replication** Default block replication is 3. The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time. Since we have 2 slave nodes we will set this value to 2.

Perform the following \$ sudo vi \$HADOOP_CONF/ hdfs-site.xml

Fill it with the following

<configuration> <property> <name>dfs.replication</name> <value>2</value> </property> <property> <name>dfs.permissions</name> <value>false</value> </property> </configuration>

Save and exit

<configuration> <property> <name>dfs.replication</name> <value>2</value> </property> <property> <name>dfs.permissions</name> <value>false</value> </property> </configuration>

mapred-site.xml

This file contains the configuration settings for MapReduce daemons; the job tracker and the task-trackers.

The mapred.job.tracker parameter is a hostname (or IP address) and port pair on which the Job Tracker listens for RPC communication. This parameter specify the location of the Job Tracker for Task Trackers and MapReduce clients.

JobTracker will be running on **master** (NameNode)

Perform the following \$ sudo vi \$HADOOP_CONF/mapred-site.xml Fill it with the following

<configuration> <property> <name>mapred.job.tracker</name> <value>hdfs://ec2-54-209-221-112.compute-1.amazonaws.com:8021</value> </property> </configuration>

Save and Exit

1.6.1 Move configuration files to Slaves

Now, we are done with hadoop xml files configuration master, lets copy the files to remaining 3 nodes using secure copy (scp)

start with SNN, if you are starting a new session, follow ssh-add as per section 1.5

from master's unix shell issue below command

\$ scp hadoop-env.sh core-site.xml hdfs-site.xml mapred-site.xml ubuntu@<URL of your Secondary Name node>:/home/ubuntu/hadoop/conf

Repeat this for slave nodes and check if they got copied in all the nodes

B ubuntu@ec2-54-209-221-112: ~/hadoop/conf					3
ubuntu@ec2-54-209-221-112:~/hadoop/conf\$ scp	hadoor	-env.sh	core-site.	xml hdfs-si	-
te.xml mapred-site.xml ubuntu@ec2-54-209-221	-47.com	upute-1.a	amazonaws.co	om:/home/ub	
untu/hadoop/conf					
hadoop-env.sh	100%	2438	2.4KB/s	00:00	
core-site.xml	100%	464	0.5KB/s	00:00	
hdfs-site.xml	100%	321	0.3KB/s	00:00	
mapred-site.xml	100%	302	0.3KB/s	00:00	
ubuntu@ec2-54-209-221-112:~/hadoop/conf\$					
ubuntu@ec2-54-209-221-112:~/hadoop/conf\$ scp	hadoor	-env.sh	core-site.	xml hdfs-si	
te.xml mapred-site.xml ubuntu@ec2-54-209-223	8-7.comp	oute-1.ar	mazonaws.com	m:/home/ubu	
ntu/hadoop/conf					
hadoop-env.sh	100%	2438	2.4KB/s	00:00	
core-site.xml	100%	464	0.5KB/s	00:00	
hdfs-site.xml	100%	321	0.3KB/s	00:00	
mapred-site.xml	100%	302	0.3KB/s	00:00	
ubuntu@ec2-54-209-221-112:~/hadoop/conf\$ scp	hadoop	-env.sh	core-site.	xml hdfs-si	
te.xml mapred-site.xml ubuntu@ec2-54-209-219	-2.comp	oute-1.ar	mazonaws.com	m:/home/ubu	
ntu/hadoop/conf					
hadoop-env.sh	100%	2438	2.4KB/s	00:00	
core-site.xml	100%	464	0.5KB/s	00:00	
hdfs-site.xml	100%	321	0.3KB/s	00:00	_
mapred-site.xml	100%	302	0.3KB/s	00:00	=
ubuntu@ec2-54-209-221-112:~/hadoop/conf\$					
					Ŧ

1.6.2 Configure Master and Slaves

Every hadoop distribution comes with master and slaves files. By default it contains one entry for localhost, we have to modify these 2 files on both "masters" (HadoopNameNode) and "slaves" (HadoopSlave1 and HadoopSlave2) machines – we have a dedicated machine for HadoopSecondaryNamdeNode.

ubuntu@ec2-54-209-221-112:~	/hadoop/conf\$ ls	
capacity-scheduler.xml	hadoop-policy.xml	slaves
configuration.xsl	hdfs-site.xml	ssl-client.xml.example
core-site.xml	log4j.properties	<pre>ssl-server.xml.example</pre>
fair-scheduler.xml	mapred-queue-acls.xml	taskcontroller.cfg
hadoop-env.sh	mapred-site.xml	task-log4j.properties
hadoop-metrics2.properties	masters	
🛃 ubuntu@ec2-54-209-221-112: ~/ha	adoop/conf	Ŷ
1 ocalhost		
~		
~		
~		
~		

1.6.3 Modify masters file on Master machine

conf/masters file defines on which machines Hadoop will start Secondary NameNodes in our multi-node cluster. In our case, there will be two machines HadoopNameNode and HadoopSecondaryNameNode

Hadoop HDFS user guide : "The secondary NameNode merges the fsimage and the edits log files periodically and keeps edits log size within a limit. It is usually run on a different machine than the primary NameNode since its memory requirements are on the same order as the primary NameNode. The secondary NameNode is started by "bin/start-dfs.sh" on the nodes specified in "conf/masters" file."

Perform the following

\$ vi \$HADOOP_CONF/masters and provide an entry for the hostename where you want to run SecondaryNameNode daemon. In our case HadoopNameNode and HadoopSecondaryNameNode



1.6.4 MODIFY THE SLAVES FILE ON MASTER MACHINE

The slaves file is used for starting DataNodes and TaskTrackers

\$ vi \$HADOOP_CONF/slaves



1.6.5 Copy masters and slaves to SecondaryNameNode

Since SecondayNameNode configuration will be same as NameNode, we need to copy master and slaves to HadoopSecondaryNameNode.

ubuntu@ec2-54-209-221-112:~/hadoop/conf\$	scp masters	slaves	ubuntu@e	c2-54-209-2
21-47.compute-1.amazonaws.com:/home/ubunt	u/hadoop/co	nf		
masters	100%	83	0.1KB/s	00:00
slaves	100%	82	0.1KB/s	00:00
ubuntu@ec2-54-209-221-112:~/hadoop/conf\$				

1.6.7 CONFIGURE MASTER AND SLAVES ON "SLAVES" NODE

Since we are configuring slaves (HadoopSlave1 & HadoopSlave2), masters file on slave machine is going to be empty

\$ vi \$HADOOP_CONF/masters



Next, update the 'slaves' file on Slave server (HadoopSlave1) with the IP address of the *slave node*. Notice that the 'slaves' file at Slave node contains only its own IP address and not of any other Data Node in the cluster.

\$ vi \$HADOOP_CONF/slaves



Similarly update masters and slaves for HadoopSlave2

1.7 Hadoop Daemon Startup

The first step to starting up your Hadoop installation is formatting the Hadoop filesystem which is implemented on top of the local filesystems of your cluster. You need to do this the first time you set up a Hadoop installation. **Do not format a running Hadoop filesystem**, this will cause all your data to be erased.

To format the namenode

Goto Namenode(master node) and perform the following

\$ hadoop namenode -format

```
221-112:~/had
 4/01/13 02:36:27 INFO namenode.NameNode: STARTUP_MSG:
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = ec2-54-209-221-112.compute-1.amazonaws.com/172.31.35.98

STARTUP_MSG: args = [-format]

STARTUP_MSG: version = 1.2.1

STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.2 -r 1503152; compiled by 'mattf' on Mon Jul 22 15:23:09 FDT

STARTUP_MSG: java = 1.7.0_45
 4/01/13 02:36:28 INFO util.GSet: Computing capacity for map BlocksMap
4/01/13 02:36:28 INFO util.GSet: VM type = 64-bit
 4/01/13 02:36:28 INFO util.GSet: 2.0% max memory = 1013645312
4/01/13 02:36:28 INFO util.GSet: capacity = 2^21 = 2097152 entries
4/01/13 02:36:28 INFO util.GSet: recommended=2097152, actual=2097152
 4/01/13 02:36:28 INFO namenode.FSNamesystem: fsOwner=ubuntu
4/01/13 02:36:28 INFO namenode.FSNamesystem: supergroup=supergroup
4/01/13 02:36:28 INFO namenode.FSNamesystem: isPermissionEnabled=false
 4/01/13 02:36:28 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
 4/01/13 02:36:28 INFO namenode.FSNamesystem: isAccessTokenEnabled-false accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
 4/01/13 02:36:28 INFO namenode.FSEditLog: dfs.namenode.edits.toleration.length = 0
4/01/13 02:36:28 INFO namenode.NameNode: Caching file names occuring more than 10 times
4/01/13 02:36:29 INFO common.Storage: Image file /home/ubuntu/hdfstmp/dfs/name/current/fsimage of size 112 bytes saved in 0 seconds.
 4/01/13 02:36:29 INFO namenode.FSEditLog: closing edit log: position=4, editlog=/home/ubuntu/hdfstmp/dfs/name/current/edits
 4/01/13 02:36:29 INFO namenode.FSEditLog: close success: truncate to 4, editlog=/home/ubuntu/hdfstmp/dfs/name/current/edits
4/01/13 02:36:30 INFO common.Storage: Storage directory /home/ubuntu/hdfstmp/dfs/name has been successfully formatted.
4/01/13 02:36:30 INFO namenode.NameNode: SHUTDOWN_MSG:
 **********
SHUTDOWN_MSG: Shutting down NameNode at ec2-54-209-221-112.compute-1.amazonaws.com/172.31.35.98
  ountu@ec2-54-209-221-112:~/hadoop/conf$
```

Lets start all hadoop daemons from HadoopNameNode

\$ cd \$HADOOP_CONF

\$ start-all.sh

This will start

• NameNode,JobTracker and SecondaryNameNode daemons on HadoopNameNode

000	ubuntu@ec2-54-209-223	1-112: ~/hadoop/conf — ssh — 1	118×31
🕲 ubuntu@ec2/conf — ssh	🔘 ubuntu@ec247: ~ - ssh	n 💿 ubuntu@ec27: ~ — ssh	O ubuntu@ec22: ~ - ssh
ubuntu@ec2-54-209-221-112:~	/hadoop/conf\$ clear		
ubuntu@ec2-54-209-221-112:~	/hadoop/conf\$ start-all.sh		
starting namenode, logging	to /home/ubuntu/hadoop/lib	exec//logs/hadoop-ubuntu-nam	enode-ec2-54-209-221-112.compute-1.
amazonaws.com.out	Franking de	transfer to theme to be at	(hadaaa Diihaaa da Daar (hadaaa da
ecz-54-209-219-2.compute-1.	-2. compute=1. amazonaws.com	tanode logging to /nome/ubunt	u/hadoop/tibexec//togs/hadoop-ubu
ec2-54-209-223-7.compute-1.	amazonaws.com: starting da	tanode logging to /home/ubunt	u/hadoop/libexec//logs/hadoop-ubu
ntu-datanode-ec2-54-209-223	-7.compute-1.amazonaws.com	.out	
The authenticity of host 'e	c2-54-209-221-112.compute-	1.amazonaws.com (172.31.35.98)	' can't be established.
ECDSA key fingerprint is f3	:90:74:77:31:5f:2f:f6:d7:5	b:94:f1:0c:65:0d:df.	
Are you sure you want to co	untu/hadoon/liberec/ /log	<pre>/ ec2-54-209-221-4/.compute-1</pre>	amazonaws.com: starting secondaryna
zonaws.com.out	uncu/naccop/ cibexec// cog	synadoop-abanca-secondar ynamen	002-202-34-205-221-47.compute-1.ama
yes			
ec2-54-209-221-112.compute-	1.amazonaws.com: Warning: /	Permanently added 'ec2-54-209-	221-112.compute-1.amazonaws.com,172
.31.35.98' (ECDSA) to the l	ist of known hosts.	cocondo nunomonado la logo ino to	(here (ubustu (hadaan () ibayan () laan
/hadoon-ubuntu-secondarynam	enode-ec2-54-209-221-112.C	ompute-1, amazonaws, com, out	/nome/ubuncu/nauoop/cibexec//cogs
starting jobtracker, loggin	g to /home/ubuntu/hadoop/l	ibexec//logs/hadoop-ubuntu-j	obtracker-ec2-54-209-221-112.comput
e-1.amazonaws.com.out			
ec2-54-209-219-2.compute-1.	amazonaws.com: starting ta	sktracker, logging to /home/ub	untu/hadoop/libexec//logs/hadoop-
ubuntu-tasktracker-ec2-54-2	09-219-2.compute-1.amazona	sktracket looping to (home/uh)	untu/hadoon/libevec/ /logs/hadoon-
ubuntu-tasktracker-ec2-54-2	09-223-7, compute-1, amazonar	ws.com.out	anca/haddop/cibexec//togs/haddop-
ubuntu@ec2-54-209-221-112:~	/hadoop/conf\$ jps		
15184 JobTracker	Contraction of the second seco	Starting all hadoon daemons fr	om HadoonNameNode
15100 SecondaryNameNode		tarting an nadoop daemons in	omnadoopramerode
15269 Jps 14857 NameNode			
ubuntu@ec2-54-209-221-112:~	/hadoop/conf\$		

SecondaryNameNode daemons on HadoopSecondaryNameNode

```
Last login: Mon Jan 13 03:15:02 2014 from bas1-malton23-1177880673.dsl.bell.ca
ubuntu@ec2-54-209-221-47:~$ ls
hadoop hadoop-1.2.1.tar.gz hdfstmp
ubuntu@ec2-54-209-221-47:~$ jps
10748 Jps
10522 SecondaryNameNode
ubuntu@ec2-54-209-221-47:~$ []
```

• and DataNode and TaskTracker daemons on slave

```
nodes HadoopSlave1 and HadoopSlave2
ubuntu@ec2-54-209-223-7:~$ jps
11816 DataNode
12211 Jps
11977 TaskTracker
ubuntu@ec2-54-209-223-7:~$
ubuntu@ec2-54-209-219-2:~$ jps
11210 Jps
10976 TaskTracker
10815 DataNode
ubuntu@ec2-54-209-219-2:~$
```

We can check the namenode status from http://ec2-54-209-221-112.compute-

1.amazonaws.com:50070/dfshealth.jsp ← ⇒ C ⊡ ec2-54-209-221-112.compute-1.amazonaws.com:50070/dfshealth.jsp

☆ ≡

NameNode 'ec2-54-209-221-112.compute-1.amazonaws.com:8020'

 Started:
 Mon Jan 13 14/23/22 UTC 2014

 Version:
 1.2.1, r1503152

 Compiled:
 Mon Jul 22 15/23/09 PDT 2013 by mattlf

 Upgrades:
 There are no upgrades in progress.

Browse the filesystem Namenode Logs

Cluster Summary

7 files and directories, 1 blocks = 8 total. Heap Size is 32.65 MB / 966.69 MB (3%)

Configured Capacity	15.75 GB	
DFS Used	56 KB	
Non DFS Used	4.14 GB	
DFS Remaining	11.61 GB	
DFS Used%	0 %	
DFS Remaining%	73.71 %	
Live Nodes	2	
Dead Nodes	0	
Decommissioning Nodes	0	
Number of Under-Replicated Blocks	0	

NameNode Storage:

Storage Directory	Туре	State	
/home/ubuntu/hdfstmp/dfs/name	IMAGE_AND_EDITS	Active	

This is Apache Hadoop release 1.2.1

Check Jobtracker status : http://<Your AMAZON MASTER URL>:50030/jobtracker.jsp

🗧 🔶 🍘 c 🗋 ec2-54-209-221-112.compute-1.amazonaws.com/50030/jobtracker.jsp							☆ :						
ec2-54-209-221-112 Hadoop Map/Reduce Administration													
Complete: Mon Jul 22 15 25 309 PD1 2013 by matt Identifier: 2014 01131425 SafeMode: OFF													
Cluster Su	mmary (Heap	o Size is 9.	31 ME	3/966.69 ME	B)	-							
Running Map Tasks	Running Reduce Tasks	Submissions	Nodes	Slots	Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes	Nodes
0	0	0	2	0	0	0	0	4	4	4.00	Q	Q	Q
Scheduling Information Queue Name State Scheduling Information default running N/A Filter (Jobid, Priority, User, Name) Example: 'user:smith 2007 will filter by 'smith' enly in the user field and 2007 in all fields													
Running Jobs													
Retired Jobs													
Local Logs Log directory, Job Tracker History													

This is Apache Hadoop release 1.2.1

Slave Node Status for HadoopSlave1 : http://<YOUR AMAZON MASTER URL>:50060/tasktracker.jsp

tracker_ec2-54-209-223-7.compute-1.amazonaws.com:127.0.0.1/127.0.0.1:39269 Task Tracker Status

Version: 1.2.1, r1503152	
Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf	
Running tasks	
	Task Attempts Status Progress Errors
Non-Running Tasks	
Task Attempts Status	
Tasks from Running Jobs	
	Task Attempts Status Progress Errors
Local Logs	
Log directory	
This is Apache Hadoop release 1.2.1	

Slave Node Status for HadoopSlave2 : http://ec2-54-209-219-2.compute-

1.amazonaws.com:50060/tasktracker.jsp

← → C 🗋 ec2-54-209-219-2.compute-1.amazonaws.com.50060/tasktracker.jsp

tracker_ec2-54-209-219-2.compute-1.amazonaws.com:127.0.0.1/127.0.0.1:48106 Task Tracker Status

Version: 1.2.1, r1503152 Compiled: Mon Jul 22 15:23:09 PDT 2013 by mattf	6
Running tasks	
	Task Attempts Status Progress Errors
Non-Running Tasks	
Task Attempts Status	
Tasks from Running Jobs	
	Task Attempts Status Frogress Errors
Local Logs	
Log directory	
This is Apache Hadoop release 1.2.1	

Running The Code:

Browse the MapReduce_Kmeans directory and locate the following files MapRedKMeans.jar, data.txt, centroids.

Step 1: Create folders in HDFS

- hadoop fs -mkdir /user/ubuntu/data
- hadoop fs -mkdir /user/ubuntu/clusters

Step 2: From PuTTy change the current directory to MapReduce_Kmeans and use the following commands to copy the files from local disk to HDFS

- hadoop fs -copyFromLocal data.txt /user/ubuntu/data
- hadoop fs -copyFromLocal centroids.txt /user/ubuntu/clusters

Step 3: Run the Hadoop jar file with following command

 hadoop jar MapRedKMeans.jar KMeans /user/ubuntu/data /user/ubuntu/clusters 3

This command will run Kmeans algorithm for three iterations.

Note: We are providing the data file and the centroids fie separately. The centroids.txt file can be generated from data file by randomly sampling. In our experiment, to keep consistency across all the platforms we select first *k* rows as centroids.

Step 4: Checking the resultant centroids after Kmeans completion. Each iteration will create a folder in HDFS named cluster*X* where *X* is the number of iteration. For example, after running Kmeans with the above command, we can view the contents with the following command

• Hadoop fs -cat /user/ubuntu/clusters/clusters2/part-r-00000

Or, we can download the resultant folder to our local disk with the following command

• Hadoop fs –copyToLocal /user/ubuntu/clusters2 /home/Ubuntu/

Stopping the Hadoop Daemon

\$ cd \$HADOOP_CONF

\$ stop-all.sh

<u>Cleanup</u> (<u>Important</u>)

Step 1: Logon to Amazon AWS and under Services select 'Ec2'.
Step 2: Under the 'Instances' tab in the left column; click on 'Instances'.
Step 3: Locate all your Hadoop instances and select them. On the top locate 'Actions' drop down button and click 'Stop' to stop the instances. You can start it and connect to the same settings whenever you want. If you terminate it, you have to create a new instance all together.

Caveats

When you stop and restart the amazon instances, the Public IP and the URL of the instances changes. You have to make changes in the following with the new URLs

- 1. hostname
- 2. Step <u>2.3.2</u>
- 3. /etc/hosts
- 4. \$HADOOP_CONF/ core-site.xml
- 5. \$HADOOP_CONF/ core-site.xml
- 6. \$HADOOP_CONF/ masters
- 7. \$HADOOP_CONF/ slaves
- 8. Repeat Step 1.5
- 9. No need to format the namenode
- 10.Start the Hadoop daemon