**Getting Familiarize with AWS**

**Creating Admin user:**

**STEP-1:** After signup with AWS login to AWS Console and setup new admin user

NOTE: Use superadmin only for billing and support purpose. (Recommended)

**STEP-2:** Goto IAM (Identity and Accesss management Console) and click on users section in the left hand side panel and create user.

**STEP-3**: Click on created user and change password in teh the Security Credentials section. And also add AdminAccessPolicy to the user.

**STEP-4:** Go to Dashboard details section you can create alias for console

**STEP-5**: Try to access URL <https://tvajjala.signin.aws.amazon.com/console>

My user: **tvajjala**

Password: **type\_your\_password**

**Creating a Key Pair:**

Amazon EC2 uses public key cryptography to encrypt and decrypt log information. Public key cryptography uses a public key to encrypt a piece of data, such as a password, and then the recipient uses the private key to decrypt the data.

The public and private keys together are known as a "key pair." To login to your instance, you must create a key pair. You specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance Linux and Unix instances have no password, rather, you use this key pair to login using SSH.

Key pairs are associated with EC2, so we'll go into the "EC2 services," and then down on the left-hand menu, we'll click on "key pairs".

**STEP-1:** Go to the EC2 section click on key pair section. And create new Keypair button to create which creates and download pem file.

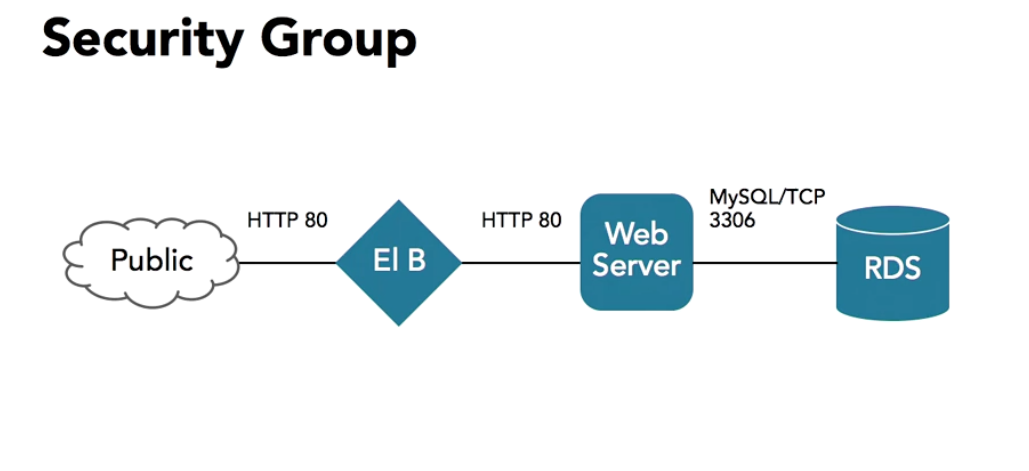
**STEP-2:** Change the permissions to this file using below command

$/> chmod 0400 dev-tvajjala-34.pem

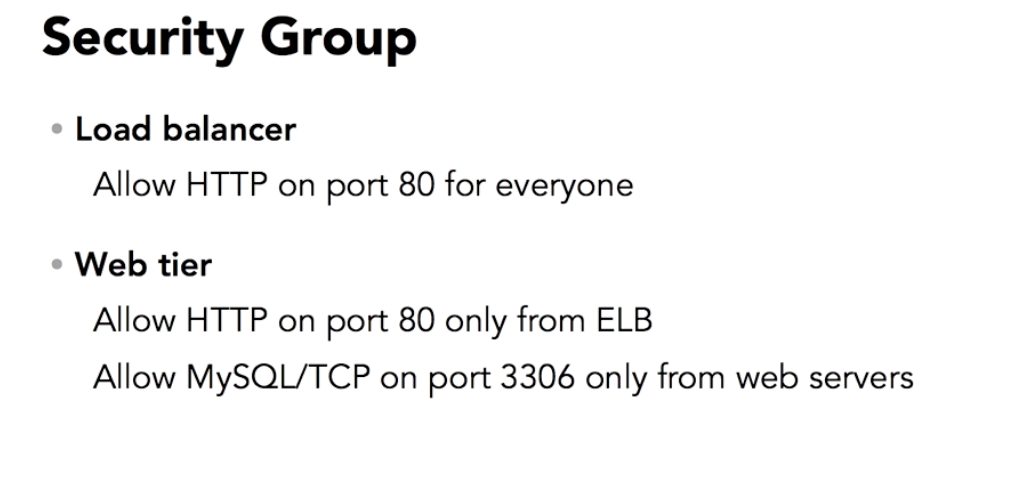
**STEP-3**: you can check permission using $/>ls –la dev-tvajjala-34.pem

**Creating Security Group**

Security groups were introduced in our chapterdedicated to keeping things secure, now let's create some. First, let's think about what needs security groups. We have public traffic coming in to our load balancer, so we want to allow for that. Then, we have the load balancer sending traffic to our web instances, so we need to allow for this, but we should restrict this access to only traffic from the load balancer.



Then, our web servers need to be able to connect to the database server, but, again, we only want to allow the connection from our web servers. So, I'm gonna create two security groups to handle this. The first one, we're gonna use for the load balancer and this one is going to allow http traffic on port 80 in from everyone. Then, we're gonna create another security group that's going to allow http traffic on port 80 but only from the ELB and also we're gonna set it up to allow MySQL/TCP trafffic on port 3306 but only from the web servers.



**STEP-1:** Go to EC2 and select Security Groups from the left hand side.

**STEP-2:** Click on create security group and provide name and description.

Choose options as shown in the below image and create new group.

Copy the Group ID from the Description: sg-647b4703

**STEP-3:** Create another security Group and add same groupId for MYSQL traffic.

**Creating ELB**

It is direct process just give name and follows the instructions on the screen.

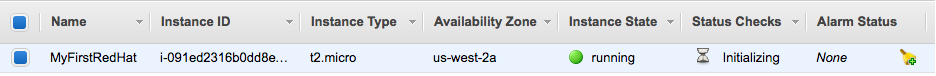
**Launching Instances**

Goto the EC2 section and click on the Launch Instance button.

**STEP-1:** Choose AMI (Amazon Machine Image) and click on Select button

Follow the steps and click on Launch button that gives creates a new instance.

Gives status of the instance as shown below when you click on view instances.



**HINT**: Amazon Elastic Block Store (Amazon EBS) provides persistent block level storage volumes for use with Amazon EC2 instances in the AWS Cloud.  Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes offer the consistent and low-latency performance needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes – all while paying a low price for only what you provision.

**Connection Via SSH**

Before this step add SSH protocol in the selected security group to so that we can connect to the instance that we created.

Use below command to change the password for user

**$/>sudo psswd**

Switching to super user

$/>sudo su -

**Getting started with YUM**

There are a variety of package managers available for different Linux distributions. Mandriva uses urpmi; Debian and Ubuntu use apt. Fedora and Red Hat use yum, while Gentoo uses portage. Some distributions provide support for more than one package manager as well.

Yum is written in python and has been in use with Fedora and Red Hat for many years. Yum has been proven to work, and despite some criticism as to its speed in comparison to other package mangers, it does the job, even if it is a little bit slower.

**YUM**(Yellowdog Updater Modified) is an open source command-line as well as graphical based package management tool for RPM (RedHat Package Manager) based Linux systems. It allows users and system administrator to easily install, update, remove or search software packages on systems. It was developed and released by Seth Vidal under GPL (General Public License) as an open source, means anyone can allowed to download and access the code to fix bugs and develop customized packages. YUM uses numerous third party repositories to install packages automatically by resolving their dependencies issues.

**What is Wget**

GNU Wget (or just Wget, formerly Geturl) is a computer program that retrieves content from web servers. It is part of the GNU Project.

Its name is derived from World Wide Web and get. It supports downloading via HTTP, HTTPS, and FTP protocols.

To display information of all file system statistics in GB (Gigabyte) use the option as ‘df -h‘.

$/>df –h

Display file system usage in MB use optio as ‘-m’

$/>df –m

Using **YUM** Package manager we can execute Commands for Linux and install software as shown below

**$/> yum update –y**

(Which updates package manager –y means don’t wait for any user inputs)

$/>**yum groupinstall –y “Web Server” “Mysql Database” “PHP Support”**

$/>**service httpd start**

$/>**chkconfig httpd on**

Once it is done try to access your machine through browser using

<http://YOUR_PUBLIC_IP_ADDRESS>

Place your files on **/var/www/html/** folder to view from browser

And Provides write permissions to the group as shown below

$:/> sudo groupadd www ( Creating group with name www)

$/>sudo usermod –a –G www ec2-user (Adding user to the group )

$/> sudo chown –R root:www /var/www ( ownership of /var/www folder to root with group www)

future files in the sub directory can be changes as shown below

$/> **sudo chmod 2775 /var/www**

$/>**find /var/www –type d –exec sudo chmod 2775 {} +**

**$/>find /var/www –type f –exec sudo chmod 0664 {} +**

**Setting up virtual machine**

**STEP-1:** Go to <http://virtualbox.org> and download the virtual box

STEP-2: Go to <http://getfedora.org> and download workstation

Follow instructions <http://www.lynda.com/Linux-tutorials/Optional-Creating-virtual-machine/435539/482229-4.html>?

**Core File Storage**

* **Warn Stoage** 🡺 this is for files that you read and write to frequently.

Simple Storage Service (S3)

* **Cold Storage**🡺 This is the type of files that are archived or that you put information into the archive and you very, very rarely will read it out.

**Apart from these two storages AWS offers two types of virtual box storages**

* Single: EBS (Elastic Block Storage)
* Multiple: EFS (Elastic File Storage)

**Exploring Simple Storage Service**

This is their core file system. So its file storage on their cloud using buckets, which are kind of analogous to drives locally and folders, which are the same thing, and then, of course, files.

Go to AWS Console and click on S3 Storage and click on blue button to create Storage.

Once you create S3 bucket you can have more options right side.

Below link gives some example bucket policies

<http://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html>

Policy can be generated using

<https://awspolicygen.s3.amazonaws.com/policygen.html>

**Logger:** This will log either all activities by default or anything you've to find via policies. So you might want to say, I want to have a log of whenever there is a delete of files in this bucket, I want to have a log ofwhenever there is a change to folder structure, so on and so forth, so all these activities can be easily logged.

**Events:** what events do is they allow notifications to be sent through another set of AWS services, so that you can then receive notifications about activities that are performed on the bucket.

Lifecycle allows you to move some of the objects from this warm storage into cold storage.

AWS Command Line tools installation can be found at

<http://mydevopsworld.blogspot.in/>

**Exploring Glacier Storage**

Storing as an archive

**CloudBerry tools**

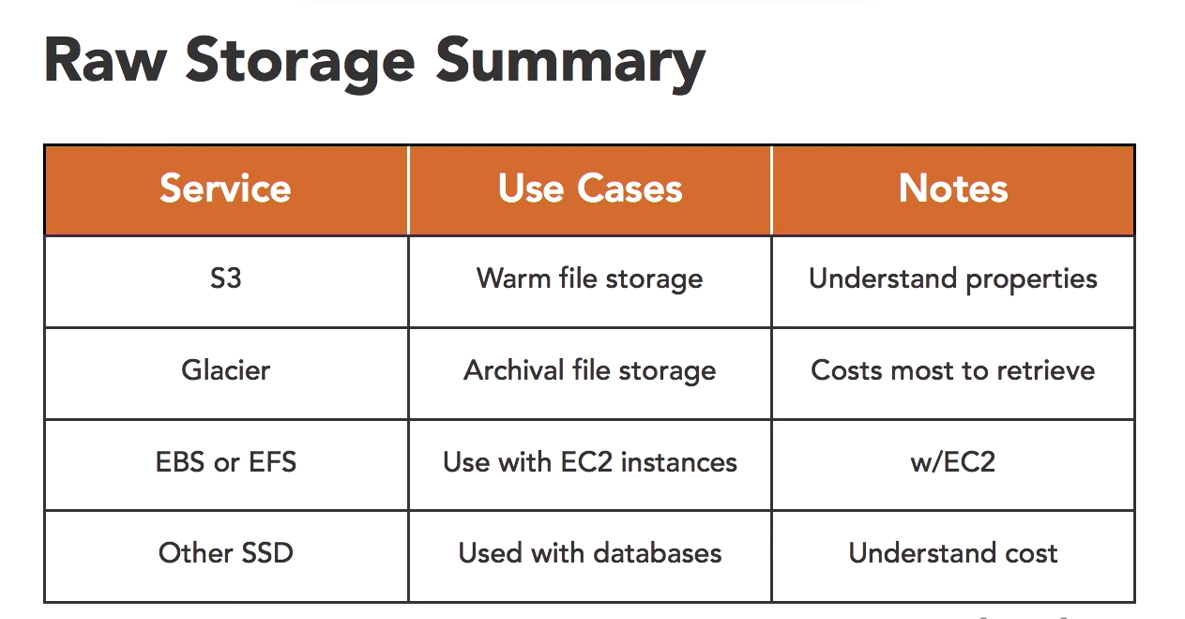
In addition to working with the Amazon console, I use partner tools in working with core file storage and archival services and one of my favorites is something called CloudBerry Lab. So they have free versions of similar tools and commercialversions, but they're relatively inexpensive and provide great value because they have a very easy, gooey interface.

You can download these tools from <http://www.cloudberrylab.com/>

**Elastic file storage**

There's a new type of storage for EC2 instances that's called Elastic File Store.

1. EBS is for one file set that's associated to one virtual machine, and
2. EFS can be associated to multiple virtual machines.



You can calculate prices using below link

<https://calculator.s3.amazonaws.com/index.html>

**Other File Storage Systems**

* **Storage Gateway**: The **AWS Storage Gateway** is a service connecting an on-premises software appliance with cloud-based **storage** to provide seamless and secure integration between an organizations's on-premises IT environment and **AWS's storage**infrastructure.
* **CloudFront:** Amazon **CloudFront** is a global content delivery network (CDN) service. It integrates with other **Amazon Web Services** products to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no minimum usage commitments.

**Exploring NoSQL Options in AWS**

**ElastiCache** is **a web service** that makes it easy to set up, manage, and scale a distributed in-memory cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution, while removing the complexity associated with deploying and managing a distributed cache environment. Is is key value store.

It's a key-value store at its core, however Amazon, interestingly, has added support for JSON data which makes it also a document database, so it's a combination of both.

It uses SSD drives, so it's really fast, it’s designed for read/write workloads, and it creates tables with keys, values, and indexes, and you set the throughput, or the IOPS, on the tables. And it's used in situations when you want to have large to huge, scalable lookups.