**AWS Linux instance connection**

Now that putty has been configured to use a proxy and a private key you can try to connect to the server hosted in the Amazon Cloud. By default, the username that you will have to use to connect to the server hosted in Amazon Cloud and to which the RSA key is associated to is the ec2-user.At the login prompt enter ec2-user for username and you will be prompted to enter the Key passphrase that you have set for the private key that you have generated with puttygen.exe. Now that you have access to the server, you may wish to set the remote access for **root.Use: sudo passwd** root to generate a new password for root, then use su root to become root and change the configuration settings for ssh.Run vi /etc/ssh/sshd\_config, set **PermitRootLogin** yes, **PasswordAuthentication** yes, save the file, run service sshd restart and from now on you should be able to login to the server using the root, without using any previous RSA key or the ec2-user username.

**Placement Groups**

**A placement group is a logical grouping of instances within a single Availability Zone.** Using placement groups with supported instance types enables applications to participate in a low-latency, 10 Gigabits per second (Gbps) network. Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. To provide the lowest latency, and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking.

## Placement Group Limitations

Placement groups have the following limitations:

* A placement group can't span multiple Availability Zones.
* The name you specify for a placement group must be unique within your AWS account.
* The following are the only instance types that you can use when you launch an instance into a placement group:
  + General purpose: m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge
  + Compute optimized: c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge | cc2.8xlarge
  + Memory optimized: cr1.8xlarge | r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | x1.32xlarge
  + Storage optimized: d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | hi1.4xlarge | hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge
  + GPU: cg1.4xlarge | g2.2xlarge | g2.8xlarge
* Not all of the instance types that can be launched into a placement group can take full advantage of the 10 Gbps network speeds provided. Instance types that support 10 Gbps network speeds are listed in the Amazon EC2 Instance Types Matrix.
* Although launching multiple instance types into a placement group is possible, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a placement group.
* You can't merge placement groups. Instead, you must terminate the instances in one placement group, and then relaunch those instances into the other placement group.
* A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs. For more information about VPC peering connections, see VPC Peering in the Amazon VPC User Guide.
* You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.
* Reserved Instances provide a capacity reservation for EC2 instances in an Availability Zone. The capacity reservation can be used by instances in a placement group that are assigned to the same Availability Zone. However, it is not possible to explicitly reserve capacity for a placement group.
* To ensure that obtainable throughput remains at 10 Gbps, members of the placement group must address each other via their private IP addresses. If members address each other using their public IP addresses, throughput drops to 5 Gbps or less.
* Network traffic to and from resources outside the placement group is limited to 5 Gbps.

AWS CLI Commands :

aws ec2 describe-regions ---To Show all regions

aws ec2 describe-availability-zones

aws ec2 describe-availability-zones --region us-west-2

aws ec2 describe-instances

aws ec2 start-instances --instance-ids i-dddddd70

aws ec2 stop-instances --instance-ids i-5c8282ed

aws ec2 terminate-instances --dry-run --instance-ids i-dddddd70

aws ec2 create-tags --resources i-dddddd70 --tags Key=Department,Value=Finance

aws ec2 describe-volumes

aws ec2 attach-volume --volume-id vol-1d5cc8cc --instance-id i-dddddd70 --device /dev/sdh

aws ec2 run-instances --dry-run --image-id ami-08111162 --count 1 --instance-type t1.micro --key-name MyKeyPair --security-groups my-ami-security-group

aws ec2 reboot-instances --instance-ids i-dddddd70

aws ec2 modify-instance-attribute --instance-id i-44a44ac3 --instance-type "{\"Value\": \"m1.small\"}"

aws ec2 create-image --instance-id i-44a44ac3 --name "Dev AMI" --description "AMI for development server"

aws ec2 describe-images --image-ids ami-2d574747

aws ec2 deregister-image --image-id ami-2d574747 && aws ec2 delete-snapshot --snapshot-id snap-4e665454

aws ec2 delete-snapshot --snapshot-id snap-4e665454

aws ec2 modify-instance-attribute --instance-id i-44a44ac3 --disable-api-termination

aws ec2 modify-instance-attribute --instance-id i-44a44ac3 --no-disable-api-termination

aws ec2 get-console-output --instance-id i-44a44ac3

aws ec2 monitor-instances --instance-ids i-44a44ac3

aws ec2 unmonitor-instances --instance-ids i-44a44ac3

aws ec2 describe-key-pairs

aws ec2 create-key-pair --key-name dev-servers

aws ec2 delete-key-pair --key-name dev-servers

**Code -> (integer)**

**The low byte represents the state. The high byte is an**

**opaque internal value and should be ignored.**

**\* "0" : "pending"**

**\* "16" : "running"**

**\* "32" : "shutting-down"**

**\* "48" : "terminated"**

**\* "64" : "stopping"**

**\* "80" : "stopped"**

**Name -> (string)**

**S3 Commands**

Synopsis : aws s3 <Command> [<Arg> ...]

## Available Commands

* [cp](http://docs.aws.amazon.com/cli/latest/reference/s3/cp.html)
* [ls](http://docs.aws.amazon.com/cli/latest/reference/s3/ls.html)
* [mb](http://docs.aws.amazon.com/cli/latest/reference/s3/mb.html)
* [mv](http://docs.aws.amazon.com/cli/latest/reference/s3/mv.html)
* [presign](http://docs.aws.amazon.com/cli/latest/reference/s3/presign.html)
* [rb](http://docs.aws.amazon.com/cli/latest/reference/s3/rb.html)
* [rm](http://docs.aws.amazon.com/cli/latest/reference/s3/rm.html)
* [sync](http://docs.aws.amazon.com/cli/latest/reference/s3/sync.html)
* [website](http://docs.aws.amazon.com/cli/latest/reference/s3/website.html)

Delecte an S3 Bucket

#aws s3 rb s3://bucket-name --force

Copy Recursively

#aws s3 cp MyFolder s3://bucket-name -- recursive [--region us-west-2]

Move S3 bucket to different location

# aws s3 sync s3://oldbucket s3://newbucket --source-region us-west-1 --region us-west-2

List the sizes of an S3 bucket and its contents

#aws s3api list-objects --bucket BUCKETNAME --output json --query "[sum(Contents[].Size), length(Contents[])]"

Display subsets of all available ec2 images

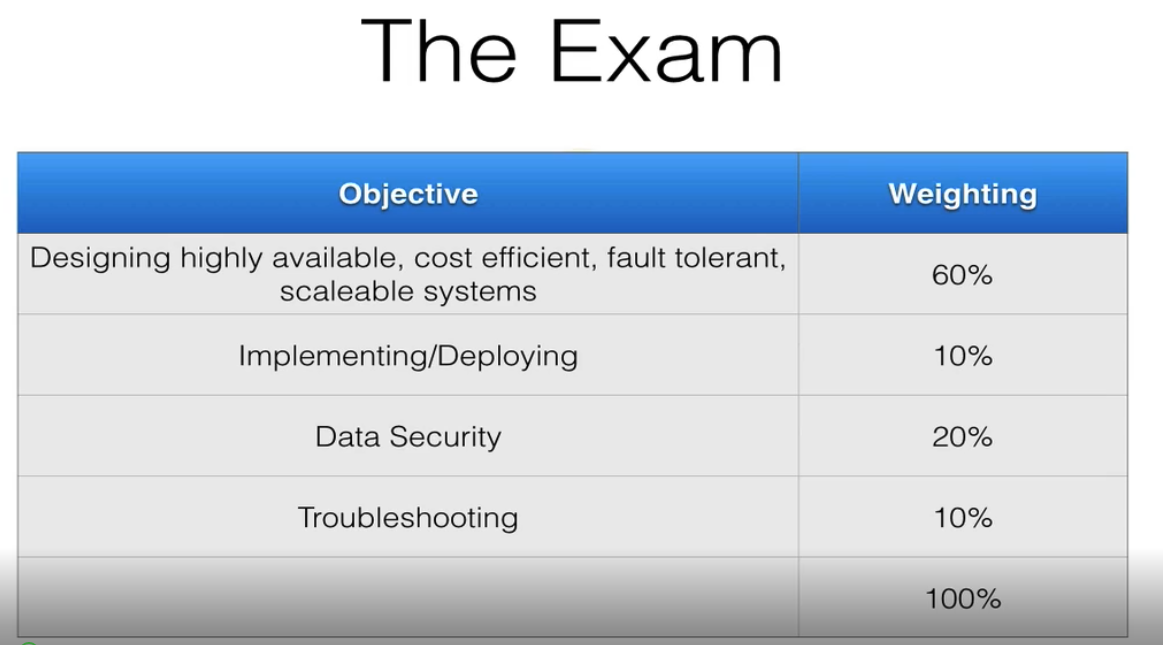
#aws ec2 describe-images | grep Ubuntu

List users in a different format

# aws iam list-users --output table

List all of your instances that are currently stopped, and the reason for the stop

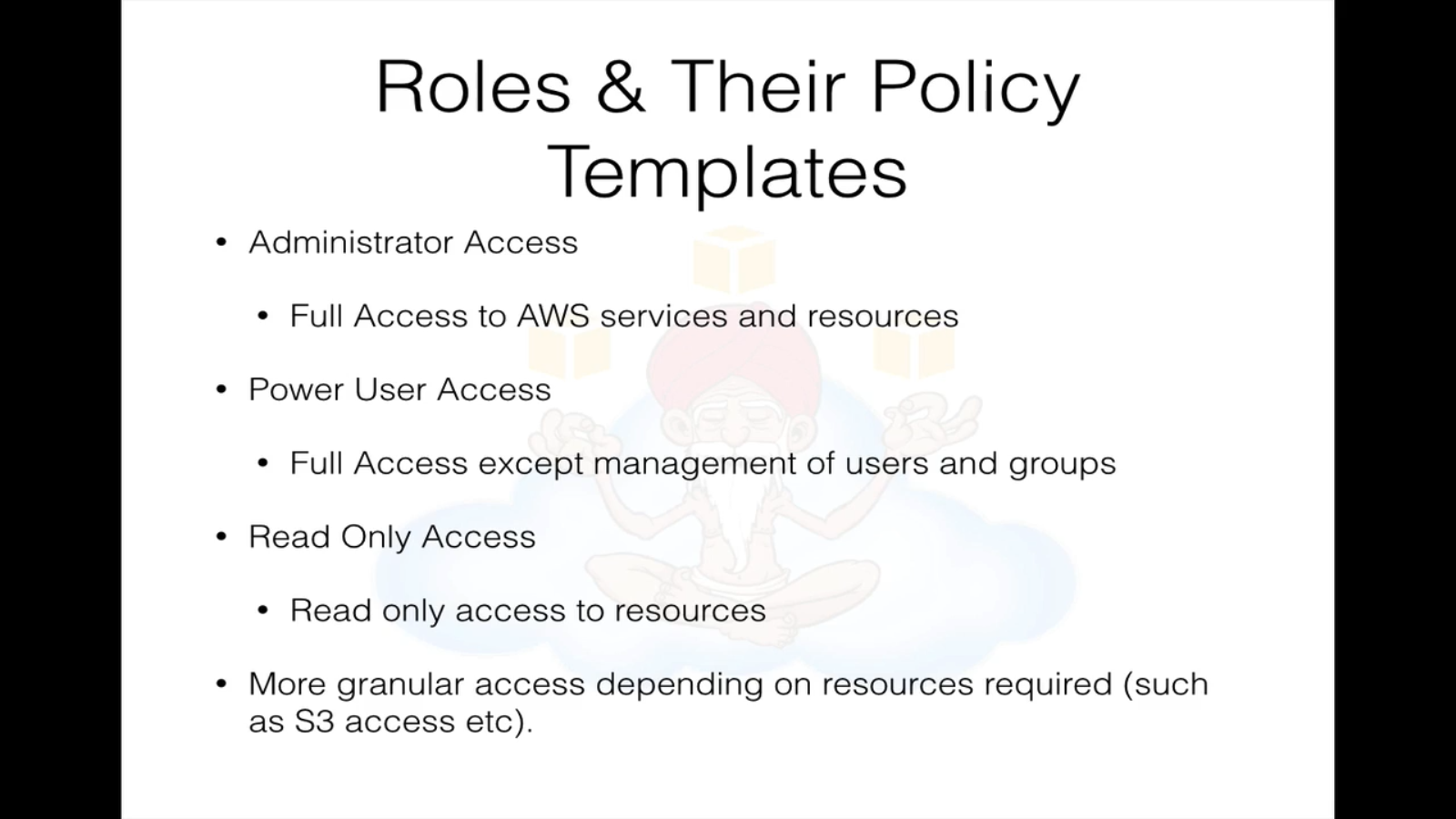
# aws ec2 describe-instances --filters Name=instance-state-name,Values=stopped --region eu-west-1 --output json | jq -r .Reservations[].Instances[].StateReason.Message



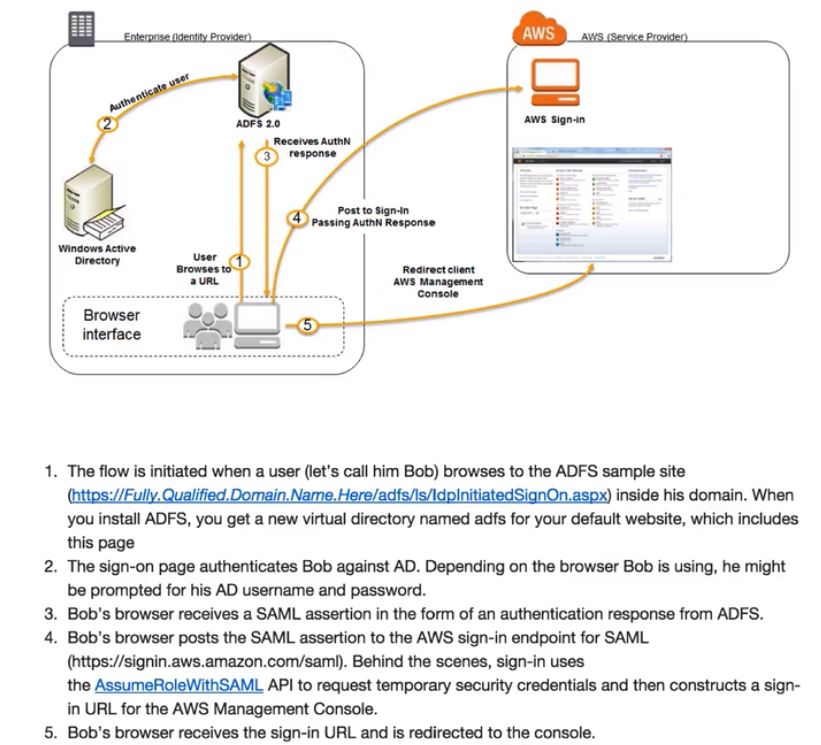
IAM:

Roles : -----🡪 Belongs users and resources

Under roles , polices are created to manage resources .



Active directory Integrate with AWS



Volumes :

# lsblk

# file –s /dev/sda

#mkfs –t ext4 /dev/sda

After mount attached volume to ec2

# mount /dev/xvda /folder

WORDPRESS:

Cronjob ----

\*2/ \* \* \* \* root aws s3 sync –recursive /var/www/html s3://bucketname

\*3/ \* \* \* \* root aws s3 sync –recursive s3://bucketname /var/www/html

Bootstrap script:

#!/bin/bash

yum update –y

aws s3 cp –recursive s3://bucketname /var/www/html

service crond start

service httpd start

UBUNTU

<https://help.ubuntu.com/community/ApacheMySQLPHP>

**S3 to ec2 instance**

## Amazon S3 and Amazon EC2

Given the benefits of Amazon S3 for storage, you may decide to use this service to store files and data sets for use with EC2 instances. There are several ways to move data to and from Amazon S3 to your instances. In addition to the examples discussed below, there are a variety of tools that people have written that you can use to access your data in Amazon S3 from your computer or your instance. Some of the common ones are discussed in the AWS forums.

If you have permission, you can copy a file to or from Amazon S3 and your instance using one of the following methods.

**GET or wget**

The **wget** utility is an HTTP and FTP client that allows you to download public objects from Amazon S3. It is installed by default in Amazon Linux and most other distributions, and available for download on Windows. To download an Amazon S3 object, use the following command, substituting the URL of the object to download.

**Copy**

[ec2-user ~]$ **wget https://*my\_bucket*.s3.amazonaws.com/*path-to-file***

This method requires that the object you request is public; if the object is not public, you receive an "ERROR 403: Forbidden" message. If you receive this error, open the Amazon S3 console and change the permissions of the object to public. For more information, see the [Amazon Simple Storage Service Developer Guide](http://docs.aws.amazon.com/AmazonS3/latest/dev/).

**AWS Command Line Interface**

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. The AWS CLI enables users to authenticate themselves and download restricted items from Amazon S3 and also to upload items. For more information, such as how to install and configure the tools, see the [AWS Command Line Interface detail page](http://aws.amazon.com/cli/).

The **aws s3 cp** command is similar to the Unix **cp** command. You can copy files from Amazon S3 to your instance, copy files from your instance to Amazon S3, and copy files from one Amazon S3 location to another.

Use the following command to copy an object from Amazon S3 to your instance.

**Copy**

[ec2-user ~]$ **aws s3 cp s3://*my\_bucket*/*my\_folder*/*my\_file.ext* *my\_copied\_file.ext***

Use the following command to copy an object from your instance back into Amazon S3.

**Copy**

[ec2-user ~]$ **aws s3 cp *my\_copied\_file.ext* s3://*my\_bucket*/*my\_folder*/*my\_file.ext***

The **aws s3 sync** command can synchronize an entire Amazon S3 bucket to a local directory location. This can be helpful for downloading a data set and keeping the local copy up-to-date with the remote set. If you have the proper permissions on the Amazon S3 bucket, you can push your local directory back up to the cloud when you are finished by reversing the source and destination locations in the command.

Use the following command to download an entire Amazon S3 bucket to a local directory on your instance.

**Copy**

[ec2-user ~]$ **aws s3 sync s3://*remote\_S3\_bucket* *local\_directory***

Configuration SSH(linux redhat

https://access.redhat.com/documentation/en-US/Red\_Hat\_Enterprise\_Linux/7/html/System\_Administrators\_Guide/s1-ssh-configuration.html