**Section 5 – Elastic Compute Cloud (EC2)**

Amazon EC2

- Most popular of AWS’ offering

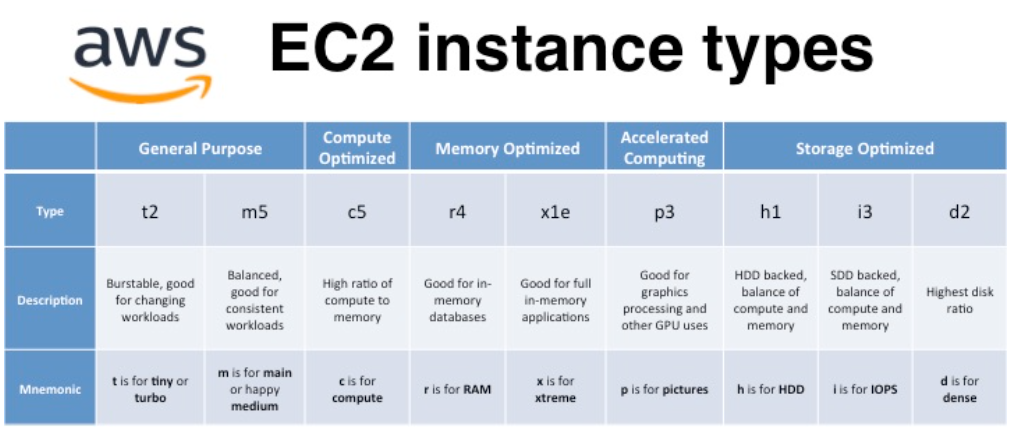
- IaaS in AWS

- Renting virtual machines, storing data, distributing load across machines, scaling the services.

Knowing EC2 is fundamental to understand how the Cloud works. Cloud is to be able to rent these compute when we need and EC2 is just that.

EC2 sizing and configuration options

* OS (Linux, Windows, MacOS)
* CPU
* RAM
* Storage space
* Firewall rules: security group
* Network card
* Bootstratp script



General Purpose

- Great for a diversity of worloads such as web servers or code repositories.

- Balance between compute, memory, networking.

- t2.micro in this course

- T, M, A

Compute Optmized

- Compute-intensive tasks

- Batch processing workloads, media transcoding, high performance web servers, computing, dedicated gaming servers.

- C

Memory Optimized

- Fast performance for workloads that process large data sets in memory;

- Use cases: high performance, in-memory databases optmized for BI, real time processing of big unstructured data.

- R (ram), X, High Memory, z1d.

Storage Optimized

- great for storage intensive tasks that require high, sequential read and write access to large data sets on local storage.

- OLTP systems, relational and NoSQL databases, cache for in memory databases, data warehousing applications, distributed file systems.

- I, D, H1.

**Security Groups and Classic Ports Overview**

Fundamental of network security in AWS.

They control how traffic is allowed into or out of our EC2 instances.

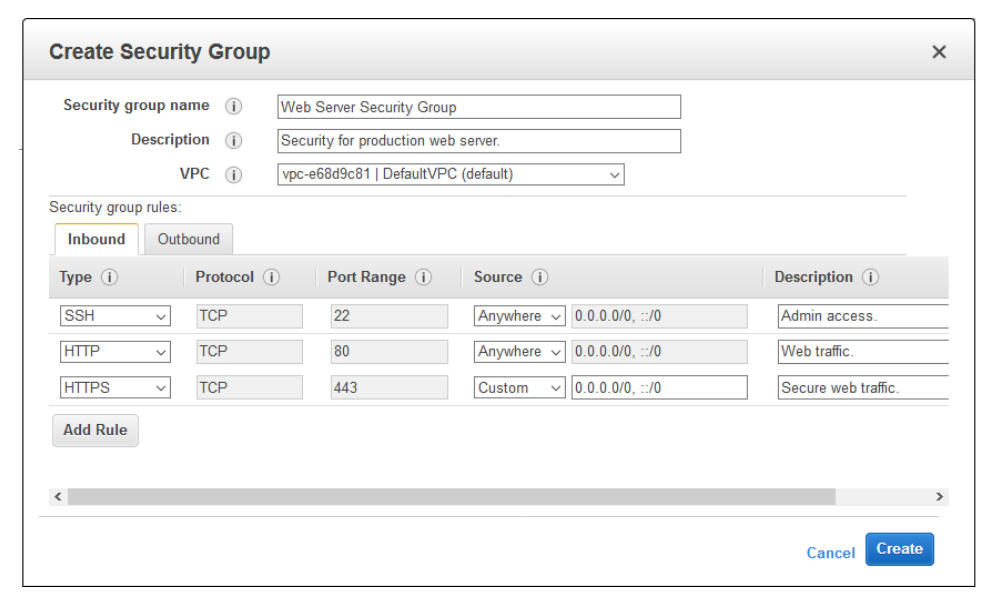
SG only contain allow rules.

SG groups rules can reference by IP or by security group.

SG acting as a firewall on EC2 instances.

Regulate: access to ports, authorised IP ranges (IPv4 and IPv6)

Control of inbound network and outbound network



- Can be attached to multiple instances

- Locked down to a region / VPC combination

- One separate security group for SSH access

- Time out: security group issue

- Connection refuses error: security group work, its and error application

- Inbound traffic is blocked by default

- Outboung traffic is authorised by default

Exam:

- 22 = SSH (Secure Shell) – log into a Linux instance

- 21 = FTP (File Transfer Protocol) – upload files into a file share

- 22 = SFTP (Secure File Transfer Protocol) – upload files using SSH

- 80 = HTTP – access unsecured websites

- 443 = HTTPS - access secured websites

- 3389 = RDP (Remote Desktop Protocol) – log into a Windows instance

**SSH Overview**

Como conectamos dentro do servidor para realizar manutenção ou ação.

Linux Servers -> SSH

Depende do SO do computador

Mac – SSH, EC2 Instance Connect

Linux – SSH, EC2 Instance Connect

Windows – PUTTY, EC2 Instance Connect

Windows>10 – SSH, PUTTY, EC2 Instance Connect

EC2 Instance Connect: usa o navegador browse para conectar a instancia EC2.

Funciona apenas com Amazon NX2.

**SSH troubleshooting** – acontece bastante problema para configurar ou executar.

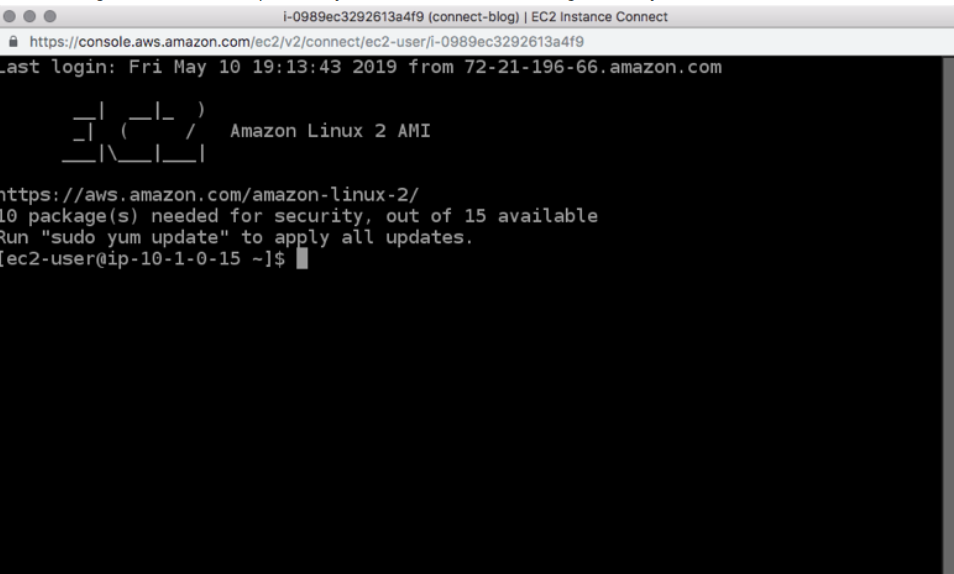
**SSH in Windows**

Allow us to control a remote machine, all using command line.

Permite que a gente controle a máquina usando comando de linha.

EC2 Instance Connect: “Connect” > EC2 Instance Connect (specify user name)

EC2 Instance



Não usa SSH key que a gente faz download manual. Usa um chave SSH temporária.

Security > Security Group

Se tirarmos o SSH inbound rules dá timeout e erros na conexão.

**EC2 Instances -- IAM Roles**

No exemplo: utilizando EC2 Instance Connect

Aws –version

Aws iam list-users

Aws configure: não recomendável (nada recomendável) rodar o comando e colocar as informações de credenciais pois qualquer outra pessoa na nossa conta pode conectar no EC2 e recuperar o valor das nossas credenciais.

Então, vamos usar IAM Roles.

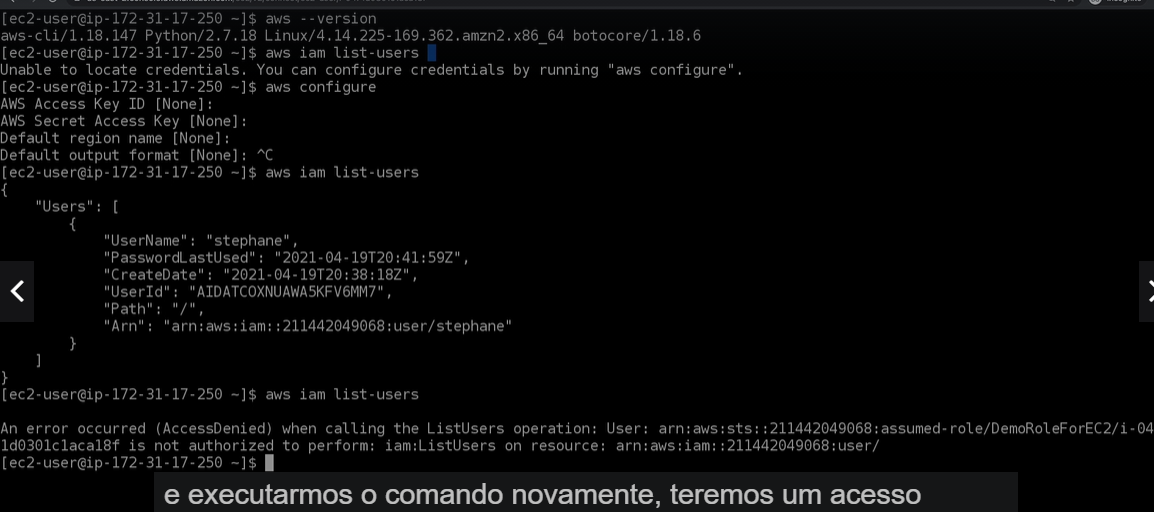
Console: IAM Roles

DemoRoleForEC2

Policy IAMReadOnlyAcces

Attach this role in our EC2 instance type.

EC2 > ACTION > SECURITY > Choose de IAM Role



Então a gente não consegue rodar aws configure e sim rodar direto o comando aws iam list-users.

**EC2 Instances Launch Types**

1. On-Demand Instances: short worload, predictable pricing

Pay what you use.

Highest cost but no upfront payment.

No long term commitment.

Recommended for short term and uninterrupted workloads, where you cant predict how the application will behave.

1. Reserved: minimum 1 year

- long workloads (ex: database)

- convertible reserved instances: long workloads with flexible instances

- scheduled reserved instances: every Thursday between 3 and 6 pm.

75% discount compared on demand, pay monthly, reserve a specific instance type

Recommend for steady state usage applications (think database)

1. Spot Instances: short workloads, cheap, can lose instances (menos confiável)
2. Dedicated Hosts: book na entire physical server, control instance placement.

Amazon Machine Image (AMI)

Has to choose the AMI: Amazon Linux 2, macOS Catalina...

Use *Quick* *Start* (AMI que já tem pronto na AWS)

>> Amazon Linux 2 AMI

Have to choose the instance type: t2.micro

Next: configure de instance details

Script: paste in User Data.

Key Pair: like na SSH key.

Rules:

- SSH

- HTTP

**EC2 Instances Purchasing Options**

1. On demand Instances: short workload, predictable pricing, pay by second

Linux or Windows: billing per second

All other OS: billing per hour

Has the hightes cost but no upfront payment

- Reserved instances (1-3 years)

Recommend for reserved instances (long workloads) and convertible reserved instances (long workloads with flexible instances)

- Savings Plans (1-3 years): more modern, commitment to na amount of usage, long workload

- Spot Instances: short workloads, cheap, can lose instances (less reliable)

- Dedicated Hosts: book and entire physical server, control instance placemente

- Dedicated Instances: no other consumer will share your hardware

- Capacity Reservations: reserve capacity in a specific AZ for any duration