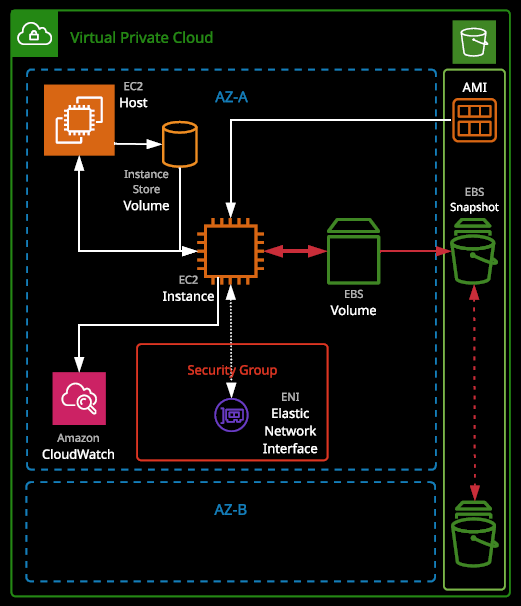
EC2 architecture.

**EC2 is a core product in the AWS line up. It's one of the first products that most people use when they're introducing themselves to AWS. It provides access to virtual servers known as instances running inside AWS with access to other AWS products and services**. So there are other companies where you can get virtual servers, but they don't have the benefit of being able to interact with other AWS features and that's what makes EC2 so powerful. Now, as I just mentioned, **EC2 instances are virtual machines, they're essentially operating systems which are running on EC2 hosts.** This is hardware that's shared between AWS customers. Now you don't see the other customers because of the advanced virtualization technology that AWS use but each instance that you create in EC2 is using a slice of these EC2 hosts. So a slice of their physical hardware. How much of a slice depends on the type and size of the instance but more on that soon.



So I'm going to be stepping through this diagram but the best way of showing you is to actually create an EC2 instance. So I'm going to do that and I'll be utilizing the Linux Academy Cloud Playgrounds. So I'm going to spin up an AWS account. So cloud Sandboxes start AWS Sandbox access. You'll need to copy the password to your clipboard. Open the Sandbox URL. Log in with Cloud\_user, which is always standard. Your unique password and click Sign In. Now EC2 is a regional service because I'm using the Linux Academy environment, I want to make sure that I've got North Virginia selected and if that's not selected, I can change it from the drop down menu because EC2 is a regional service, it means that any resources that I create in one region will be isolated to that region. So it's especially important if you are creating any EC2 instance to avoid any bill shock in your own accounts make sure you remember the region where you creating them but of course, if you're using the sandbox that Linux Academy provide, you don't need to worry about that because it's all included in the cost of the subscription. Now to get started, I'm going to go to the EC2 console before I create an EC2 instance, I'll need to create a **key pair**. As the name suggests, it's actually **a pair of keys a public part and a private part and these will be used to authenticate to the EC2 instance,** So I'll create a key pair. A key pair needs a name, so I'll just call it EC2 for now and hit Create. Now what happens is that two parts of this key are generated. **The public part is stored by AWS and installed on any EC2 instances when you choose to use this key, the private part the PEM part is downloaded this time only to your machine. This is the only time you can ever get this private key part,** so it's essential that you keep it safe because it's what you'll use to connect to the instance. If you ever lose this, you won't be able to use this key again, you'll need to remove the key, recreate it, and then obviously handle any existing instances which use the old key pair and there is a process that you can go through to replace that, but I'll talk about that later in the course. Now there's one more thing that I need to do. This is an EC2 lesson.

**EC2 instances do run inside and networking environment in AWS called VPC, or virtual private cloud.** Now you need to make sure when you're creating an EC2 instance that it has a network to operate in because I'm using a Linux Academy provided account that is created without any infrastructure, I need to create a VPC and it needs to be a VPC that's already set up by AWS. So I'm going to do that. I'm going to move across by clicking the services dropdown, typing VPC, and I'm going to create a default VPC. The default VPC is essentially just a prebuilt VPC created by AWS automatically and it includes all of the necessary networking infrastructure. In the next section of the course, I'll be talking in depth about networking, but in advance of that, we're just going to go ahead and use the default VPC. So click your VPCs, go to actions, and then create default VPC, ht create again. It'll take a couple of seconds, hit close, and you'll be able to verify that it's the default VPC by scrolling to the right and checking the yes in default VPC. That's all we need to do AWS have preconfigured that for us now. So I'm going to move back to the EC2 console. Now that's the prep steps, so I've gone ahead and created the networking infrastructure and we've created our SSH key pair to log in to, so I'm going to go ahead and click on launch instance. The first thing that you'll need to do **when creating an EC2 instance is to select the AMI or Amazon machine image.** For now, I want you to think about **this is the installation media for the EC2 instance. It contains all of the information required to build your EC2 instance, so to install the operating system on the system or the root volume. So for now, just think about it as installation**. The only thing you'll need to do is to select the operating system that you want to use. So there's various different distributions of Linux available. The top one is Amazon Linux 2 which is version two of the Linux distribution that AWS make. There's also Red Hat, SUSE, Ubuntu, versions of Windows that you can use if you decide on. So you can always go ahead and select community AMIs or search for a particular Linux distribution but it's worth drawing your attention to the AWS Marketplace because it's here where you can also find lots of commercial AMIs. For now, though, I'm just going to select Amazon Linux 2 to keep things simple. The one thing that you do get to decide on in selecting the AMI to use is whether to pick the ARM or the X86 architecture. Now, these are just different types of CPU is most instances in AWS use Intel CPU so X86 but you are able to select ARM CPUs which is generally lower power and more suitable for certain types of workload. For now, though, I'm going to select X 6 and hit Select. Now at this point, you'll have to select the type of instance that you want to create and this is influenced by the AMI architecture. So because I selected X86 I have got a large range of instance types to select from. Had I selected ARM I would only have a subset. Now I'll talk about instance sizes and types more in an upcoming lesson but be aware that the size and type that you select will influence the amount of virtual CPUs that you have, the amount of memory, as well as the speeds of the storage and networking and other capabilities, so you need to pick the type of instance and select it very carefully from this list.

Don't worry. I'll have a dedicated lesson coming up in this topic of the course, which will help you select the appropriate instance for a given situation. For now, I'm just going to pick T3.micro, which includes two CPUs, 1 GB of RAM, and it's a nice, cheap balance of resources. It's good enough for a demo, so I'm going to select that, and then I'm going to click Next Configure Instance Details to move on. Now it's on this screen that you'll select various important instance options, the first two of which relate to the networking. Every **EC2 instance belongs to a specific VPC or virtual private cloud, and more so it lives in a specific subnet in a specific availability zone of that VPC** so you need to select both of those. Now because we're using a default VPC, I'm going to leave this set to default, under subnet, I'll click on that dropdown, and I'll be able to select either no preference, which will pick a subnet out of all the ones that are available. I can, though, select a specific subnet to use, which uses a specific availability zone. So because you U.S. East North Virginia has a lot of different availability zones, I've got a lot of different subnets to pick from. It's important to understand that an **EC2 instance lives in a specific availability zone because if an availability zone fails, then the instances in that availability zone will also fail and so for larger architectures, you're generally going to be provisioning things in lots of different availability zones.** I just want you aware at this point that an EC2 instances inside a single availability zone. So I'm going to pick U.S. East 1a for now, it should automatically select to use subnet setting enable for public IPs. So this will receive a public IP. So I can go ahead and accept all of the rest of the defaults at this stage. Now, don't worry a lot of these are important, and I am going to be covering them in dedicated lessons later in this topic and future topics. For now, I don't want to add any unnecessary complexity, so I'm going to move past these. I'm going to go to Next Add Storage.

**Now an EC2 instance, is capable of using two different types of storage: instance store volumes are physical storage devices that are attached to the EC2 host that an instance runs on. With certain types of instances, I'm able to add these instance store volumes.** It's important you understand the limitations and capabilities of these instance store volumes and I'll be covering that in detail in a future lesson in this topic but just to introduce **them instance store volumes are directly physically attached to a single EC2 host and so if the device fails, then you will lose any data on that instance store volume. There's no resilience. Additionally, if the EC2 instance has to move between physical hosts either because the host fails or that you do a stop and start of the EC2 instance which moves it to a new host, then you lose all of the data on the instance store volume because you'll be moving to a new host with new volumes. Instance store volumes are not persistent. The data is not secure. It could be lost at any time and so they're general used for temporary high performance requirements. Instance store volumes tend to be the highest performance of any storage type**, and again, I'll be covering that in detail in a future lesson in this topic, **the other type of volume available are EBS volumes or elastic block store. These are provided by another service called EBS. You create the volume in EBS and attach it to an EC2 instance, and that works over the network. EBS is capable of providing lots of different types of volume using different types of drive to provide those volumes** and again, I'll be covering those in depth in the future lesson in this topic. For now, just realize that **based on the AMI that I select, it will provide a default and this default is 8 GB of general purpose SSD storage, or GP2.** There are lots of different ways you can change this, and I'll talk about them in detail in a dedicated lesson but for now, I'll be able to accept the defaults and move on to tags.

**Tags are a way that you can apply key value pairs to AWS resources. A fairly common usage for EC2 instances is to add a key value pair where the key is name and then the value is the name of the instance.** So in this case, EC2 demo, but you could easily add additional key value pairs, which might link this instance to a specific cost center or a specific platform type such as prod or development or staging. Tags give you a lot of flexibility, and you can use them for almost anything that you can imagine but for now, I'm just going to use this simple pair of name and then EC2 demo. I'm going to click next to Configure Security Group. Now, **security groups you can think of at this stage as a virtual firewall**. I'll have a dedicated lesson coming up later in this topic, but **essentially they allow you to filter the traffic that can come in to an EC2 instance or leave an EC2 instance**. Now, security groups are presented as being attached to an EC2 instance. That's not strictly true. **When you create an EC2 instance, you're creating a default ENI or elastic network interface, essentially, the instance's network card. The security group is attached to this network interface and because it's the default network interface it's presented as the security group is attached to the instance.** Now, by default, the security group that you create along with the instance will allow the default protocol. So if it's a Linux instance, it'll allow SSH which is port 22 from anywhere. If its a Windows instance, it'll allow RDP. Now, I always recommend changing this from custom to my IP so that only you can connect to this instance. It just adds a little bit more security. I also recommend that you change the security group name and description to something a little bit more descriptive. So instead of launch wizard, I'm going to call this EC2 demo, and I'm going to use that for both the security group name and description and then click on Review and Launch. Everything looks good. Click launch. I'm going to choose an existing key pair. I'll select key pair that I created at the start of this demo, acknowledge that I still possess it, and launch the instance, and that a high level is how you can create an EC2 instance. Now I will be covering a lot of the surrounding technologies in detail as we move through the fundamentals topic of this course but I just wanted to provide you with a high level overview of how you create an EC2 instance and attempt to give you the key architectural facts as I went through the process. Now this lesson is getting a little bit on the long side, so I wanted to split it into a number of different parts and give you the opportunity to take a small break. So this is the end of part one. Go ahead and mark this lesson as complete and when you're ready, you can join me in part two.