Getting Started
Speaker: Dave Rocamora, Amazon Web Services Solutions Architect

>>: Welcome to today's webinar, Getting Started with AWS. Our presenter today is Dave Rocamora. Dave is a solutions architect with Amazon Web Services. He's based in Seattle, Washington, and on a team that supports AWS's global network of technology consulting partners. Dave works to help partners and their customers build setting things with AWS. Dave's background is in consulting on technology systems for media and entertainment, financial services, and casino gambling.

Today with us we also have Matt Lewis, solutions architect; Thomas Robinson, solutions architect; Ian Scofield, solutions architect; and, David Potes, partner solutions architect. They will be our webinar moderators and they'll be engaging with you and answering your questions throughout presentation while our presenter presents.

Dave, welcome. The floor is now yours.

>> DAVE ROCAMORA: Hi, everyone. This is Dave Rocamora. I'm a solutions architect at AWS. I work with customers and partners to help them build solutions on top of AWS and take advantage of AWS technology. Thanks for joining the webinar today. It will be about getting started with AWS. So kind of some things that you can do to take your first steps when working with Amazon Web Services.

So, let's just go over what we'll get -- what we'll cover today. So what we're going to start with is creating an AWS account. So I'll walk you through how to set up an AWS account, where to go, and how to get that going. Next we'll start with some of the very first steps that I think are important to do on any AWS account, like creating an identity and access management user, and enabling multifactor authentication to secure your account so that you start with a secure environment to work with on AWS.

Next I'll go over launching and connecting to EC2 instances. EC2 is the Elastic Compute Cloud. As a service, it allows you to launch virtual machines for service in the cloud. Once we have those EC2 instances up and running, I'll cover how to back up and restore those instances so that you can save them for later; and then we'll take a look at the S3, or the Simple Storage Service, to store and serve files over the Internet.

Next we can take a look at the -- how to visualize AWS costs and set billing alerts so that you can understand exactly what you spend might be inside of an AWS account.

So, you know, before we dive into the hands-on stuff, I like to go over this -- the characteristics of what cloud computing is and sort of the tenets that AWS follows. So, cloud computing is the on-demand delivery of I.T. resources over the Internet with pay-as-you-go pricing; and I think it's important to understand these things because these themes will repeat themselves through all the services that we're working with.

So what do we mean by on-demand delivery? That means that, you know, when you're requesting the resources from AWS, spinning up servers or creating storage or something like that, you're getting, you know, what you need when you need it. It looks like my slides shifted a little bit here. Okay. I think we're good.

So you're getting what -- the resources that you're requesting, you get them as soon as you ask for them. There's no calling us up and prepositioning things or building a data center or ordering servers. You know, things are delivered on-demand.

When we say I.T. resources, we mean anything that's involved in I.T. So, you know, compute resources like servers, storage, networking components, and higher order things like databases or application services like queues. Things like that. They're all delivered to customers over the Internet, and they have pay as you go pricing. So everything that you use is a provision in a pay as you go model, like the utility based pricing model, as you might -- just similar to how you might buy electric or water from the power company or the water company. So for what you use, you might be paying for compute hour or per gigabyte per month for storage or something like that.

Okay. So another thing that I want to point out to people that are just getting started with AWS is that there's a free tier. So the free tier includes most AWS services and it's available for all new accounts. The free tier is good for one year from the day the account is created. So if you're creating an account today, you can utilize free tier for up to a year; and that's for some services.

For other services, there's a perpetually free tier thing going on there where you get so much free every single month. And it's a really -- the free tier is a great way to get -- to understand, you know, how to use AWS and explore how to use AWS without making some kind of financial commitment to it.

I want to point out that everything that we are showing you can be done within the free tier today. So everything today you can do within the free tier; and if you want more details about what's covered under the free tier and what's not, or, you know, some information about that, this website, aws.Amazon.com/free has a lot of detailed information and the fine print on what's available in the free tier.

Okay. So the first thing that we're going to do and the first thing you need to do to work with AWS would be to sign up for the -- sign up for an AWS account. So you can go to the website aws.Amazon.com. You will need to provide some information, including a credit card for some -- if you go over the free tier and to be able to charge it to that, and there will also be a telephone verification. So that's a process where an automated system will call you to prove that you're a real human, you know, to make sure -- to help you set up your account. The process takes, you know, in ten -- you know, five or ten minutes. So it's very easy to get started and do that.

Okay. So let's do a quick demo. I'll show you on my screen, you know, what it might be like to sign up for an AWS account. Okay. So first while we're here, right, this is the website about the free tier, aws.Amazon.com/free; and you can see on this page more detailed information about what services are covered under the free tier and what kind of configurations and, you know, for how long. So, good thing to check out.

Next you're going to go to sign in. I mean, here we are on the main page, aws.Amazon.com. Signing in is really straightforward, or signing up is really straightforward. Just click on this button, "Create an AWS Account", here; and then you'll have to provide some information. You can put your e-mail address in to here. You could say that you're a new user, and sign up. It will be pretty similar to signing up for any other kind of service. You know, you have to put some information in, including your credit card and a telephone number. The telephone verification happens, and then you'll get an e-mail a few minutes later and you'll have -- you'll have an AWS account.

There's something else I want to show you while we're here in this demo queue, this getting started center. So, you know, in this webinar today, I'm going to go through a couple of things -- how to launch instances, how to configure S3 -- but if you want to do something at your own pace, you know, on your own, or you want to go deeper on a specific area, we have a website called getting started at aws.Amazon.com/gettingstarted. And it's a lot of self-paced, ten minute tutorials about how to do certain things, like launching instances or con figuring certain kind of databases, things like that. This is a good way to sort of get your feet wet. Now, a lot of these things are covered under the free tier, too, so another good way to explore AWS.

Okay. And I think we can go back to the slides now. And let's see. So, you know, what do I do -- you know, customers often ask me, you know, what do we do within an AWS account when we first sign up for the account? Like, you know, now that we have an account up and running, what's the first thing to do? So security I think is the most important thing at AWS and the most important thing that you can do in your account.

So the first thing that I like to do is create individual users, identity and access management users. So when you have your account, when you sign up for your account, you'll get a log-in that will have your e-mail that you sign up for the account with and a password. We call that account the root account. It has the capability to do all kinds of stuff within your AWS account. It can create any kind of resource in the account, it can change your billing information, and all kinds of stuff like that.

For me, that's too much power for day-to-day stuff; so, I tend to create an IAM user first. So an IAM -- IAM stands for Identity and Access Management. It's a system that allows you to create and manage groups of AWS users and groups of those users. You can control with a very fine grain detail, you know, what kind of services people have access to. So perhaps you want to give an account that only has access to work with EC2 but not with S3 or vice versa; or maybe you want to give someone an account that allows them to look at the billing information, but not able to launch anything in the services. You can do that with IAM.

Another thing that's really important, too, to do with your AWS account when you're first signing up is to name a multi-factor authentication. So, multi-factor authentication is a process by which you provide at least one extra factor of authentication when you're logging into an account. So right now, so normally we would log in with your user name and your password. Multi-factor authentication would have an extra factor of authentication, so something extra that you need.

The way that AWS implements this is with tokens or codes. So you might be familiar with, like, this is a cute little icon of a -- like, a Gemalto token, which provides a little one time code that's synced with your account. So every time you log-in, you'll have to, you know, push that -- you push that button and get a number, and you can type it in. You can get those hardware tokens just like this to associate with your IAM users and your root account or you could use a software token through an application on your phone, which is a good way to secure your account.

This protects against things where someone, you know, gets your password. So you write it down on a post-it note -- hopefully you're not doing that but maybe that happened -- or you leave your laptop somewhere and someone gets access to it. Having the second factor of authentication can protect your account and all of the data inside of it. So what I would strongly recommend would be to enable multi-factor authentication on the root account and every single identity access management user you create within your account.

Okay. So we can go back to the demo now and go through the sort of the first steps here. So what we're going to do when we're setting up the -- those IAM users under that root account. So -- let's see. Here we go. So this right here, this is the AWS console. So here I am. I'm logged into the account, you know, for the first time; and you can see all of the different services that are available. There's over 50 services here that you can access. We're only going to focus on a handful of them today.

But so to start setting up those IAM users, what we want to do is go to the identity and access management console. So you click over here, and this should load up. So, we have a whole bunch of different things that we can do in here. The first -- like I said, first what we want to do is set up the users so that we can access the account securely without using the root account. So we'll click on "users" and there's already a user in here, but let's make new one for the demo here. So we would click this button that says "create new user," and then we can give it a name. Demo sounds like a pretty reasonable name for this account.

Something to note here on this page, we have the ability to generate an access key for each user. So AWS, you can interface it -- interface with all of it through this console, but there's also a set of APIs that you can use, third party tools or write software to interface with AWS. Access keys are the credentials that an API client might use to access AWS. It's -- if you're going to need one, it's a good idea to generate one right now. So why not? So we can say "Create" Here.

Okay. And now here's something critical to keep in mind when you're creating users in AWS or in Identity and Access Management. The credentials for that account are only shown to you once. AWS doesn't store or manage those credentials for you, you know, ongoing. So you're going to need to store them and keep track of them. This is so that someone can't gain access to things in their account that you're not -- they're not supposed to have that access to. So this is your one chance to, you know, show these security credentials or download them.

So this is what those API credentials look like. They're useful for plugging into a configuration file or a settings file or a tool that might access the API. You can also download them here, and I downloaded a csv file that I can use to access the account. So that's the last time we'll see those credentials; so, make sure to keep track of them.

Okay, great. So now that we have that user, we'll click on the demo user here; and what we can see is we can see some information about the user. So, you know, there's some basic details, you know, about what the user is and it has this permissions tab, but let's look at this security credentials tab first.

So under here we can see the access key. So this is the access key, the API credential that was associated with that user and some information about it. This is just the ID of the key, not the secret part of that key.

If we want to use this account on the console, we're going to need a password to log into the console. So what we'll do is we'll say "Manage Password" here, and we have a couple of options. We can set our own password, we can require -- or we can allow AWS to generate a password for them, or we can just require a new -- and we can also require a new password at sign-in. So let's just do that right now. So I'll set up an account and password here; and then once again, this is the last time we would see that password. So if we showed that there, we would see this password that we could use to log into the account. Okay. So we can close that.

It's also reminding me here, hey, by the way, this is the last time you're ever going to see those credentials. So, do you want to download them?

Okay. So now that we have a password for the account, to enhance the security we're going to want to turn on multi-factor authentication. So for that we're in the same section here. We're in the security credentials tab of the user, and we can go down to multi-factor authentication and manage and click "Manage MFA Device."

So if you have a hardware token sitting around right now that you're not using, you could associate it with this; but for most of us a mobile phone is probably more likely what we have on hand. So we could set up a virtual MFA device, and you can use an application like Google Authenticator on Android to do this. So if we click through this, what'll happen is there's a -- if you've done multi-factor authentication before, this is probably familiar to you. There's this QR code that you can scan with your device, and then enter the codes that it produces into the -- into this window. And then MFA will be setup. I'm not going to go through all of the setting up MFA for this account, but I would strongly recommend that you do that.

Okay. So now we have, you know, our password set, our access key set, and we can log into the account, on to the console or the API with that, and we have MFA turned on. Great. We're all done, right? Well, not quite. So by default, IAM is secure by design. So this user doesn't have access to do anything within AWS. So we need to add some permissions for the user.

So click over on this permissions tab, and then here we can attach policies. So there's two different ways you can do policies. You can do manage policies, which are policies that are built by AWS, or you can do inline policies where you can customize and write your own policies and attach it specifically to that user.

I like to use a managed policy because I think it's just a lot easier to handle the basics here. So over here, once we click that tab, I can see all these different kinds of policies that are associated with the account that we could use, and there's all different kinds of things that you could do that are prebuilt policies by AWS to apply to any different kind of use case.

The two most common ones that people use for this -- for their powerful IAM user accounts is this administrator access policy right here. This would give someone the ability to do pretty much anything inside of the AWS account. It's almost equivalent to the root account. Or the power user account, so we'll look up that. Power user access is a good choice for your -- for your main working IAM user in the account because it gives you access to everything except for IAMs. So you can do anything except for create new users and groups. So I think it's a good choice. I'm not going to pick that right now, but this is where I would -- this -- you would select one of these to associate it with the account, with this account to be able to access that there.

Okay. So now that we've done that, we have an IAM user that has access to do things in the account. So now we have what we need to move forward and start, you know, creating stuff inside of the account. So we can go back to the slides real quick here and move on to the next step.

All right. So now we have users in the account. We've done sort of the basics, you know, around setting up -- around setting up access to the account. We probably want to do something in the account, and the first thing that people tend to like to do, and I think this is a good thing to start with, is to launch an EC2 instance. So EC2 is the Elastic Compute Cloud. It allows you to create virtual servers on top of -- inside of AWS.

So, there are a couple prerequisites we need to go through to be able to launch EC2 instances; and they're mostly around being able to connect to and access the instances. So the way that we're going to do this is by creating an SSH key pair. So SSH stands for the secure shell, if you haven't heard of it before, and it's a way to remotely access computers. If you've used a Linux or Unix computer before, you're probably familiar with SSH.

We use the keys to access -- a lot of customers to access EC2 instances. SSH keys are -- consist of -- key pairs consist of a public key and a private key. You store the -- you keep the private key on your own laptop or your computer someplace, and then the public key is positioned on the EC2 instances to allow you to access them.

The SSH keys avoid password weaknesses through -- they're sort of stronger than passwords, they're harder to guess, they're longer. You know, so it's more protection. You can import your own key or have AWS generate a key for you if you want, and I'm also going to point out, AWS doesn't store the private part of the key at all. So this is another thing that you are going to have to keep track of and store to be able to access your instances that you launch inside the account. This also means that AWS can't access what's running inside of your EC2 instances. That's up to you to do, and AWS won't be accessing that or working with the stuff that's inside of your EC2 instances.

All right. So let's do a demo here and go and create a key pair. Okay. So we're back here in my AWS console, and let's click over to EC2. So EC2 is the service that allows creating virtual servers in the cloud.

Now, something I want to point out, you know, right away inside of the -- inside of EC2 is that, you know, in AWS, we have concepts of regions. And a region is a collection of resources located in a specific geographic area; and regions have really become really important with services like EC2, because you want to know, like, where your things are running and understand that, you know, things that are running in one region aren't necessarily replicated or running into another region.

So this little drop-down up here on the right shows all the different regions that are available to me, and you can see they're located all around the world. You might want to pick a region because you want to launch servers or resources closer to your customers. You also might want to pick a specific region because you have data sovereignty concerns. Like, you have data that must be stored in the European Union. You might want to choose the Ireland or Frankfurt region to store that data. So you have your choice there.

AWS won't move your data between regions but a kind of common thing that I see sometimes with new customers is that they'll be working and working and setting something up and they'll go back in the console and it looks like everything is gone, but what's really happened is they're just looking in a different region. So pay special attention to what region you're working in when you start. I'll choose Oregon because I'm here in Seattle and it's pretty close to me, but you can choose whichever region you think is best for what you do.

Okay. So now with that out of the way, let's create that key pair. So over here on the left-hand side, we can see this big long list of different things that we can do inside of EC2. Let's pick "Key Pair." So in "Key Pair" here, there's already a key pair that exists, but let's create a new one right now. We can also call it the key pair demo, and what it'll do is create that key pair and add it to the list here.

And what it did was it downloaded this -- downloaded that pem file. That's the private key. That's something super important. Don't lose that. Without that, you won't be able to access the EC2 instances you launch with this key pair. So you want to keep track of that. If you already have an SSH key that you want to use, you know, for EC2, as well, you can hit "Import Key Pair" here. You know, either one a fine.

Okay. Cool. We can go back to the slides now and we'll move on to the next step of things that we'll need to do to access -- to launch an EC2 instance. Okay. So now we have -- the key pair allows us to -- like, it's the authentication that allows us to log in; but to be able to get to the EC2 instance, we're going to be able to, like, access it over the Internet, over the network.

So every single EC2 instance is sort of surrounded by a firewall that we call a security group. By default, a security group will block all traffic into the instance. So you won't be able to use anything to connect to the instance, and it's up to you to choose which ports and protocols to open. So you can use port ranges. Like, so you could pick a specific port like, you know, 22 or 3389 or, you know, 1234 -- whichever port you want to pick. And you can also pick ranges if you want to open up a range of ports -- 8,000 to 9,000 for some reason, you can do that.

And you also get to choose who the ports are open to, and you can do this by specifying networks or groups of IP addresses using CIDR notation. So if you're not familiar with that, that'll look like an IP address and then a slash and then a number. Just sort of like a click -- a quick tip. If you're not into networking, you know, so much, a /32 is a single IP address. If you know your IP address on the Internet, you can use that /32 to, like, target you directly.

You can also create rules that specify security groups for other EC2 instances. So let's say that you're launching a bunch of web servers, and you want to allow them to connect to a database server. You can have a database security group that trusts the web server's security group and allows access on certain ports of that.

So let's create a security group that allows us to access these -- to allow us to access EC2 instances that we launched. So we go back here to the console here, and we can see a demo of this. Okay. So back in the EC2 console, we can click over on "Security Groups" here on the left. Let me get this out of the way down here. Okay.

So this is a couple of different security groups in here, and let's just accurate a new one. So we can give this thing a name and a description -- you know, whatever we want -- and then we get to define the rules. I usually like to pick something actually descriptive. So this way if I'm looking at this later, I can understand, oh, yeah, this is what I was supposed to be doing with this group.

Okay. So then we can add rules to the security group here. So we can add -- let's add a new rule. So when you're adding a rule, you're specifying a protocol and a port range, and then the source of where the traffic is coming from that you're allowing. When it comes to the -- in this -- in the console, we give you a couple of sort of shortcuts to different protocols that are very popular, customers like to use. So we're going to access the Linux instance. We'll pick SSH. So that'll just pre-populate with the TCP protocol on port 22.

When it comes to where we want to allow access from, we have a couple different choices here. I can type in a specific IP address or a specific range that I know about using CIDR notation, or I could pick a couple other things. We could do this anywhere, which provides this CIDR range of 0.0.0.0/0, which means anywhere in the entire Internet. That would open up to the entire world.

Sometimes this is appropriate, but this is probably not the best choice for allowing remote access to your computers. So a better idea would be to do from an IP address specifically to you. So the console -- so if you just pick this "My IP" choice, the console will make a best effort attempt to figure out what your IP address is and log it down, and you can see it's this big long number with a /32, which means it's just this IP. The console usually does a pretty good job of this, but you might want to double check with your ISP or with your network to make sure that it gets that right. You could always change these later, so if you want to see if your IP address changes, you want to make a rule -- the group more permissive, you can always make changes to this.

If we're going to access Windows instances, too, you know, we're not going to be able to just use the SSH. We might want one to use remote desktop to access a windows computer. So in this list, we could make another rule for that. So let's see if we can find that in here. A-ha, here we go. So RDP is another protocol that we could use to access Windows instances running inside of EC2. And so this prepopulated, you know, the TCP protocol, port 3389; and once again we can pick different IP addresses from -- let's just pick my IP. Okay.

Great. So we'll create that security group, and there it is. We can then use that group to be able to allow remote access into the -- into the EC2 instances that we launch into that group later on. Okay. So now we can go back to the slides here, and we'll move on to the next steps.

Okay. So now really what we've done is done all the prerequisites, all of the things that we need to do to be able to launch an EC2 instance. So in instances of virtual server or virtual machine that's running inside of the cloud, I want to point out once again that you have full control over what's going on in the instance. You can install any software that you'd like. So that includes customizing the software with -- you know, you could add different users, you could install the applications that you like to use, and you can configure it in any way they that you like.

You can choose the instance, type, and size to get different kinds of configuration out of the computer. So if you have a workload that needs a lot of CPU, you can launch an instance type that provides more CPU power. If you need more memory, that's a choice, too; and you can sort of customize it all all over the place. There's a list of -- we'll go through this in the console. There's a group of -- there's set instance types and different sizes in each type so that you can get an instance type that matches exactly what you're -- what you want to use. You can always change that stuff later, too; so, don't feel like you're making a commitment to a certain instance-type.

When you launch an EC2 instance, you'll need to specify a key pair and a security group to launch the instance, too. So, fortunately, we just created two of those; so, we should have everything we need to launch a new instance inside of EC2. So let's go back to the console and take a look at that, launching an EC2 instance.

Okay. So we can just go to the instance tab, the instance section over here on the left; or we can go all the way back to the dashboard. This is the page that you would land on if you clicked on the EC2 service in the console. This "Launch Instance" button is what we can use to launch it. Now, once again I'll point out, make sure that you know what region you're in. So we're in Oregon still, which is good because that's where we created the security group and the key pair, but always good to check in my opinion.

Let's click "Launch Instance" here. So now, you know, the first step that happens in the launch instance wizard, this is what this is called, is we get to select the -- the Amazon Machine Image or I call it AMI. Some people say AMI. You know, either way seems fine to me.

So this is where you pick the operating system, the base image of what's going to be on that computer. There's a ton of different options here and different ways. So under the quick start tab, you can see some pretty typical choices -- Amazon Linux, Red Hat Linux. If we scroll down a little bit, we can also see Windows instance types.

There's also the ability over here, you can get AMIs that you create yourself. So if you've made an image of a server that you want to reuse or the marketplace if you want to use commercial software inside of your AWS account, as well as community AMIs. So these are your AMIs that are published by, you know, other people that wanted to share their great machine image. I typically pick one from the quick start section. And so for today, we'll quick pick -- I'll probably pick this Amazon Linux one here.

Something else to take a look at if you're trying to explore things under the free tier, make sure you're picking an AMI that's eligible for the free tier. You'll see them kind of clearly noted here inside of the console. If you're not picking one of those, there might be some extra charges incurred that won't be covered by the free tier that would cover things like software licensing. So, look out for that if you're trying to stay within the free tier.

So pick Amazon Linux here. Okay. Next we get to choose the EC2 instance-type. So this is what's the configuration that will dictate how much CPU power and how much memory and how much disk performance and other special features that an instance might have. So, scrolling through this list, you can see a pretty long list of different kinds of instance types that are good for different kinds of things. You can see the general purpose instance family, these m4s. There's also computer optimized instances, c4s, and you can see the big differences has to do with the amount of -- scroll up here -- the amount of CPUs they have and the amount of memory they have available.

So, you know, there's -- when you order servers, like physical servers to install or set up a real computer, like, a physical computer, you might be thinking that you're making a commitment; so, you want to buy a server that you might grow into over time. But with EC2, you don't have to do that. You can pick -- you can pick whatever type you like and then change it whenever you want. It's as easy as stopping an instance, changing the type, and starting it again. So it's pretty typical to explore and try things out here, and don't stress out too much about picking the right instance type at first.

If you want to play in the free tier, the t2 micro is a great -- is the type that's covered under the free tier, and it's actually a pretty good instance type. It has variable capacity. It's good for exploring, experimenting, and doing some things in there. It's a pretty worthwhile instance to try out. So let's go with the t2 micro.

Next, we get to configure the instance details. So this is a lot of maybe more advanced things that you might be able to do within EC2, sort of exposes some of those features. The one that I'll point out here quite specifically would be this assigning of the public IP. So to be able to connect to this instance over the Internet, we're going to need to have a public IP associated with it.

So, you have a couple of different choices here. You could use what the subnet says, which is the configuration of your PC. If you just leave it as the default, you will get a public IP; but you may also choose to enable or disable, depending on what you're doing. So if you are doing this on your own, make sure that you're assigning a public IP to the instances that you're launching. Otherwise you will not be able to connect to them.

Okay. So we'll go through there. Next we can add some storage. I'm going to circle back and cover storage a little bit later, but just know that you have the ability to customize the amount of storage in the instance at launch right here, so.

Next we can tag the instances. So tags are metadata that's associated with an EC2 instance that we can use to identity the instance or report on it or something like that later. A typical first tag that everybody creates is the name tag. So this allows us to easily identify the instance when we're looking at it in a console or in other places. So, you know, let's just make an instance called "My Demo Instance." That's what the name of this instance will be.

Next we're going to configure a security group. So this wizard is going to -- it could guide us in creating a new security group here, right here and right now, or we could select an existing security group, and we could pick the one that we just created. So we'll pick this demo security group. That's the one that I just created a couple minutes ago.

Okay. Now we come to this sort of review page where we get to look at all the choices we made, make sure that we have everything right, that we're using the t2 micro instance, that we're launching the right AMI, et cetera. If everything looks good here, you can click "Launch," and this allows us to select the key pair that we want to use to access the EC2 instance.

So we can -- you can choose an existing key pair here to pick the one that we already created, or you can even just create one right here. I would not recommend proceeding without a key pair because without that you will not be able to access the instance. So once you have the key pair and you know that you have that private key file, you can say I acknowledge that I have access to this and launch the instance.

Okay. So if we go back, so now the instance is launching. So let's go back over to the EC2 console and take a look at what that looks like. So inside of here, you see this "My Demo Instance." That's the instance we just configured and launched, and it's up and -- it's starting inside of EC2.

Now, just like in a cooking show, I'm just going to sort of cut to the chase while we wait for it to launch. Usually it takes, like, one to two minutes to start up and be ready to access, but sort of to save time on the webinar here, I've already launched two instances in the account, a Linux instance and a Windows instance. So we can see what it's like to access those two.

So let's take a look at the Linux instance first. So, the Linux instance is up and running. There's some information about, you know, how it's configured and what it's doing. Some relevant information to be able to connect to it would be to understand what it's public IP or DNS name is. We'll copy this -- I'm going to copy this to my clipboard. And then to understand what key pair. So I launched it with this key pair could rocamora-demo.

Okay. So then to access this instance, what we'll need to do is use SSH. So if you're on a Mac or a Linux computer, SSH is built into the computer under the terminal program. If you're on Windows, you're going to be wanting to use a program like PuTTY, for example, to access that. The EC2 getting started guide, EC2 -- getting started with EC2 guides have information about how to configure PuTTY to use the key pairs that you generate in the console. So it might be worth taking a look at that if you're connecting from Windows; but if you're connecting from Linux or Mac, what you want to do is use SSH.

So we'll use SSH and what we need to do is specify the key pair file with this -i flag, which says, like, hey, use this private key to connect to the instance; and then specify a user name. On Amazon Linux and most, most Linuxes, the AMIs, the users' EC2 user is the default user on there; but that might be different based on your AMI. So why not take a look at that? So, the EC2 user at the IP address or the DNS name of the instance.

So if we do like that, and we're logged in. So now this is the Linux instance. We can create -- we can create files, we can configure this instance however we want. We could install software. Let's install a web server right now, so this becomes a web server. Okay. So we can log in and customize this instance. This is ours to do whatever we need to do with it. So, you know, we have that all set up.

So what does that process look like if you are trying to connect to a Windows instance? So let's take a look at that. So this Windows instance, we can set this up -- I had launched this earlier, and it's launched in the same kind of thing. It has an IP address, a DNS name; it's in a security group that allows access; and it has the key pair associated with it.

And you don't have SSH a Windows instance. You would use RDP to connect to that, but how do you get the password to connect over RDP? So, if you launch -- when you launch that EC2 instance and specify the key pair, what it did was it -- EC2 provides a method for retrieving that password. So what we can do is go into this actions menu with the Windows instance selected. We can click this button, says "Get Windows Password." And so what this does is it starts a process where it uses the private key file that I have to pull down and decrypt the Windows password.

So let's choose this file. We'll need to provide the private key. This is the private key I used to launch this instance. It goes in here, and I'll say "Decrypt Password." And so that will provide, you know, the information. So it got the -- it's printing out here the DNS name for this instance, the user name, and the password that can be used to access it.

So once we have that information, we can take that to a remote desktop program. So this is RDP on my Mac. This is installed on your Windows computers as a terminal service client, and I put that information inside of here. So from there, we can then say "Connect." "Continue." And now we're connected to the Windows EC2 instance that's running. You know, we can customize this and install stuff on this, you know, in any way that we, you know, might want to.

Okay. So -- yeah. So now that we're up and running, I'm going to go back to this Linux instance in a minute; and I'll just sort of refresh what I did here. So I installed some software and I customize this instance to make it into a web server, let's say; and I loaded my application, whatever I wanted to do there.

So we go back to the slides and talk a little bit about storage on EC2 instances. So just back to the slides real quick and we'll talk about the storage.

Okay. Cool. So, sort of flew through this when we were talking about launching the instances; but the -- but storage is, you know, obviously pretty important. It's where you're going to store your data and the stuff that's running on your EC2 instances. So the storage service for EC2 is called Amazon EBS, the Elastic Block Store. It provides persistent block storage for EC2 instances. So you might be familiar with things like this if you worked with a SAN or a NAS or even having a hard drive in your own computer, something like that.

With EBS, you can create volumes that are as small as a gigabyte and as large at 16 terabytes to store all kinds of data. They're available in several different types and I'll go through those in a second here and also, you can create snapshots of EBS volumes to create -- to make backups. So once you've configured -- once you've started storing data on EBS, you'll probably want a make a backup of it. Snapshots are a great way to do that. And we'll walk through that in just a second here.

But first, you know, a lot of customers ask me, you know, what are the best -- you know, what choices should I make in EBS similar to, like, picking the right instance type. There are a couple of different options. There are two different flavors of solid state disks, the general purpose or gp2 and provision IOPS SSDs. There's also a couple different types now of magnetic storage. They're called the throughput optimized HDD and cold HDD. All of these have different characteristics and are good for different kinds of workloads.

Typically, what I would say the best choice for almost every workload, at least getting started, would be to use the general purpose SSD, the gp2. Let's put a star next to it because this is really your -- this is -- this fits most use cases. When you know that you need to use a different type, you can certainly use a different type; but typically I would start with gp2. It provides the best balance between price and performance.

It's also really easy to change this later. So if you're running a gp2 because you followed my advice and then a couple months down the line you find that you need more IOPS out of it, you can create a snapshot of that volume and create an IOPS, a provisioned IOPS volume based on that snapshot. So you're not making a commitment to a certain EBS volume type. Gp2 is really the best choice and if you click through the wizard like we just did, it'll default to gp2. So that's typically your best choice.

All right. So let's go through just a quick demo on that so you can understand, you know, some kinds of things that you might want to do with that. So what we'll do is we'll make some backups and some images of the instance that we customized here. So this is that Linux instance that we customized. To get to

EBS, it's inside of the EC2 console under volumes here and -- thanks for that notification there -- and we can see all the EBS volumes that are launched and running inside the account, and there -- some of these are attached to different instances.

So we'll just pick one of these, and you can see -- you can get information about this volume. This is the page where you could create new volumes if you wanted to -- if we just want to make a backup, a point in time backup, we could say create snapshot here. Call this the demo snap, and create -- say "Create." That'll kick off a process that will make a copy of that -- a snapshot of that volume that we can create a new volume from later. So if you want to make backups of things in EC2 that are stored data on EBS, snapshot's a very good way to do that.

If we want to make a copy of the whole instance -- so if we wanted to relaunch that instance with the same configuration, we can do that by creating our own AMI. So we talked a little bit about AMIs at the beginning; but, you know, now that we have this instance that's set up and configured, if we wanted to be able to launch one and save all the work that we did into the image, what we can do is create an AMI from the running instance.

So to do that, you go to this "Actions" page here; and what you can do is say -- actually -- sorry. Under this image -- under the image menu item -- sorry, I lost it there for a second -- you can say create image, and so this takes you to this wizard that allows you to build a new AMI. So you can provide some details, like a name and a description. You can customize the storage on it or change the storage size or whatever you wanted to do, and what that'll do is build a new AMI that you can do when you go through the launch wizard again to be able to customize the -- to be able to have a precustomized instance that launches with your web server and all kinds of stuff are running on it. So very useful way to build, like, reproduced full servers inside of EC2.

Okay. Great. So that's storage and snapshots and AMIs on EC2. Let's go back to the slides real quick and we'll talk about another service here. So we'll just go back -- okay. So we'll talk about Amazon S3. So S3 is the Simple Storage Service. It's actually one of the oldest AWS services. It's just about -- it just turned ten years old very recently. It's an object storage service that allows you to store, you know, almost any kind of data -- almost any amount of data. You can store objects that are as small as one byte, as large as five terabytes, and you can store as many of those as you want. So a lot of storage, basically.

S3 is designed to be highly scalable and very durable. So S3 can scale to support any size, you know, a number of very large objects; but it also can scale as far as serving those objects to loss lots of customers. So if you have lots of customers downloading objects from S3 or you have lots of systems doing, say, scientific processing and pulling and pushing data to S3, S3 will scale to meet those requirements.

S3 is also very durable. When you store an object in S3, it's replicated to multiple data centers within a single region, within a region of AWS and it's constantly checked and it's checked to make sure that it's quite available and quite durable. So if you store something in S3, you can kind of trust that it's going to be there when you need to get it back out.

So encryption is available. So if you want to store sensitive data in S3, that's totally an option. You can encrypt it. There's many different kinds of encryption. There's AWS provided encryption with AWS-managed encryption keys or customer-managed encryption keys. Sometimes customers might choose to just encrypt data before they even store it to S3. That's totally fine, too.

The objects that you store in S3 are going to be in the region that you choose. So if you want to locate objects in a region closer to your customers for latency reasons or if you want to locate them in a region that has certain compliance requirements, you can do that. We won't move the data for you.

So with the objects you store there, you can set permissions on them. So if you want to make one public or one very private or make it possible for certain kinds of people to be able to download them, but not certain -- but not others, you can do that there. And as a result, it's very accessible and it's a pretty straightforward API. You can access it directly over the Internet; and in a demo, I'll show you that.

There's also lots of third party software that's designed -- that plugs right into S3. So, for example, using some backup software today, like maybe CloudBerry backup. It's -- it ties directly into S3 to store your backups in there. So you might be surprised. You might be working with some software that will use S3 natively to store data.

Okay. So let's go through a quick demo here about storing and serving some objects from S3. So back here in the console, we'll go back to the main page here. So this is once again all the services. Let's take a -- let's focus on S3, the storage service in the cloud.

So the top level organizational item inside of S3 is called a bucket. I already have one bucket in here, but we can create -- let's create a new one. You'll need to have a bucket to be able to store any objects in S3. There's a limit to the number of buckets you can create in your account, and, you know, sometimes, you know, new customers start thinking, I'll make a new bucket for every single thing that I'm doing, you know, or for every single customer that I have or something like that. That's typically not the way to go. You sort of kind of want to limit the number of buckets that you have. Inside of the bucket, you can use the fine grained permissioning controls to sort of main space the bucket and divide it up.

When you create a bucket, you can choose which region it's in. So this is where all the data stored in that bucket will be geographically in the world. Once again, I'll pick Oregon because it's close to me. This doesn't really affect how the data is available around the world, it just affects where it's stored and where it's coming from. So you might want to pick -- as a reminder, you'll pick a region based on maybe the latency to your customers and also if you have some kind of data sovereignty concerns about the data stored in there.

So let's try to create a bucket and we'll call it demo and see what happens here. So click the create, and we're going to get an error message back. So one thing that the customers will run into when they're new to S3 is that the name space for S3 buckets, like the names of the S3 buckets, is shared globally. So pretty much everybody has to sort of cooperate. The demo bucket is not available because someone else got it before me. Surprise.

So it's good to pick a unique name. Sometimes I'm almost surprised by what names are available and what names aren't but what's a good -- what a good choice to do is to, you know, put some unique information inside of the bucket name. So "webinardemo-20160426." That's today, and we'll see if that gets created.

Okay. So we've created this bucket. So now we have a place to store objects into S3. So if we click inside this bucket, we can see that it's empty right now. There's nothing -- there's no objects stored inside of it. So let's start by uploading an object. So there's lots of different ways that you can upload objects to S3. For today we'll just use the console here.

So we'll click this upload button and then we'll add some files. So I have a file here that I would like to upload, and we'll just select that and say "Start Upload." So you can use all sorts of tools to do the uploads. You know, the command -- there's a great Command Line interface. There's third party software that'll interface directly with S3. The console is also fine for some kind of things.

All right. So now we have the objects stored in here. We can take a look at it. Let's take a look at the properties of this object. So you can see it has some information, some -- size, you know, who set it up, things like that. And it has a hyperlink to the object.

Let's try to open it up and see what happens here. So, you try to open it up and it said that the access to that file was denied. So, things that you store in S3 are private by default. So if you're storing -- so it's safe to store things in here like backups or sensitive information or whatever it might be. They're not going to be public, you know, by default. You have to explicitly make that public.

Now, this is a photograph that I want to share with all of you; so, why not make it public? So once again, we're in the S3 console here under properties for this object. Let's go down to permissions. And here's where we can change permission. So I can grant very specific permissions to allow, you know, someone else in this account or someone within another account or someone in a specific IP range to access this object; but we'll just open it up to the world. So we'll grant everyone the ability to open and download this object. So we'll say "Save" here. Okay.

So now when we go to this link, we should see a picture. Okay. And we do. So this is a photograph I took a couple years ago, a tilt shift in Brooklyn New York. But yes, and now this object is being stored from the -- stored in this S3 bucket and served to the world here.

So as you can -- so S3 is pretty flexible. Some people are doing some really interesting stuff with it. In fact, you can serve -- you might realize, you know, we're basically serving an image over the Internet here that could be part of a website; and you might sort of think, what could you do with S3 like that? And, in fact, some people serve entire websites through S3. It's quite capable of doing that. So if you want to serve a static website or ask us to support your static website, you can do that directly over S3 which is pretty useful.

Okay. So that's storing and serving things in S3. Let's go back to the slide, and we'll cover one more thing here before we open up to some Q&A.

Okay. So now that we've set up a whole bunch of stuff on our AWS account -- we have EC2 instances with -- we have IAM users, we have S3 buckets. We're trying to stay under the free tier if we can; but, you know, how do we know that we're in there and how much we're spending? Remember, with AWS, you're doing pay-as-you-go. So you're paying for exactly what you're using, but it's a good idea to keep track of what you're running.

So once you have some stuff up and running, it might be good to take a look at some of the billing and go cost management features and I'll show a few here. So there's a bunch of different ways you can use to help you monitor and manage your costs, tools like Cost Explorer and alerts, which I'll show you in a minute here. We also have detailed billing reports that get you an Excel -- like a csv file that you can load into Excel to understand exactly where you're spending your money, as well as consolidated billing. So if you have a lot of different AWS accounts that you're organization use, you can use consolidated billing to roll that all up into one account so that you're only paying one AWS bill.

But, you know, for getting started and, you know, with your first AWS account, I think it's worthwhile just to look at the Cost Explorer and to look at setting billing alerts. So we can set a billing alert to understand how much we're spending on our AWS account. So let's just go back to the console, and we'll do one quick demo of that.

All right. So billing is in a little bit of a different section in your AWS account. It's -- if I click under my name here, this is the IAM user that I use to log into this account. We can go to "Billing & Cost Management." So the cost --

billing and cost management console has a bunch of different interesting stuff inside of it.

This is a really brand new AWS account, so it actually hasn't generated any bills yet. It was created just the other day. But you can get sort of a breakdown of, you know, what you're spending your money on and where it is. Cost Explorer is another way to do that. Once again, I don't believe there's an update in here for anything to show up, but if I wanted to understand how my EC2 spending was changing over time, Cost Explorer is a really useful tool to doing that.

Certainly something you can do proactively to make sure that, you know, we're -- we know how things are going inside of our AWS account and set up a billing alert. So why don't we set up an alert that says, you know, if my AWS bill goes above $5, please send me an e-mail. Send me an e-mail so I can do something about that. Either celebrate because my website is super popular or panic and log in and take a look at what's going on inside of there.

So to do that in the billing console, you might want to go down to this "Preferences" section and turn on receive billing alerts, and then from there we can click on this button that says "Manage Billing Alerts." So this brings us to the CloudWatch console. I didn't talk about this service directly, but CloudWatch is a service that allows you to monitor metrics that are coming from things in your account, including bills.

So, I already have a billing alarm set up in here; but we'll just create a new one here. And -- total estimated charges, and I check here. So this is the total estimated charges. You know, we can look -- we can see, you know, this would show as a graph of what we're spending, you know, over time in here; and we could set this up. We can -- we could have -- we could configure it all on here.

So if you're doing it for the first time, you have a window that allows you to create an alarm that just looks like this one here, where you say "Create Alarm"; and you can say $5 as your upper bound of that metric. And you can add a notification with an e-mail address here. So if we go to "Modify," this is what it'll look like.

So we'll set up a -- we'll set this up again. This is what the wizard looks like. So I say if my charges exceed $5, we put our e-mail address in here; and you can say, you know, "Save

Changes," and that'll set up a billing alert. And it'll send me a notification to my e-mail that I can use to set up a confirmation of that, and then what'll happen is I'll, you know, if the account ever goes above -- the estimated charges ever goes above $5, I'll get an e-mail to say that that's happened, and I can log in and take a look at what's going on inside of the account.

Okay. Well, that's all the -- so now that you have that set up, your account is pretty much configured and ready to go on AWS. You've done a couple things inside of here, and you've set up billing alarms so that you can understand and monitor what's going on inside of your account. You know, with that, thanks for joining the webinar today. I really appreciate everybody's time.