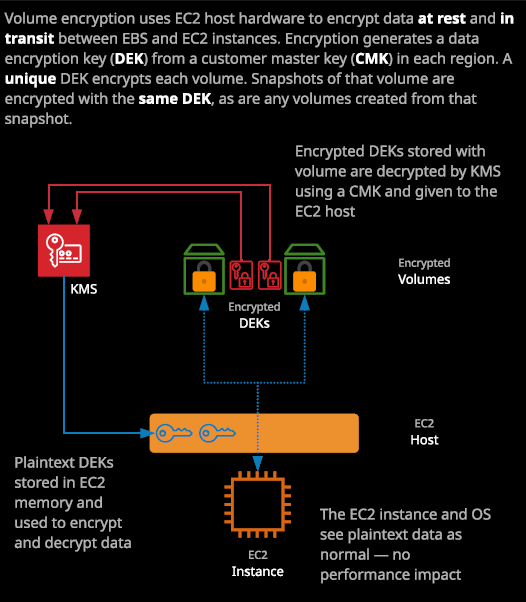
EBS volume and snapshot encryption



Now volume encryption uses EC2 host hardware to encrypt data at rest and in transit and that's between the EBS volumes and the EC2 instances. So, essentially, any time you use an EC2 instance with volume encryption, you know that any time that data is stored on a physical disk it's encrypted and that data is encrypted as it transits between the physical disk and the EC2 instance. So you've got both of the different types of encryption covered in transit and at rest.

EBS encryption is something that's configured on a per volume basis, but you are able to set it from an account perspective, EC2 console > settings > control default volume encryption settings for this entire region. So enabling this ensures that all new EBS volumes in the region that you're currently located in will use encryption.

Now, using encrypted volumes is fairly easy, and I'll demonstrate that quickly by creating an EC2 instance. So I go to launch instance, I'll pick my usual Amazon Linux 2, I'll select a T3.micro. Click on next to configure instance details. All of these look good. I'll make sure it's got a public IP. I'll click Next Add Storage because I've changed the account default for this region to be encryption on it is enabled by default when I'm creating this new volume, this dropdown will show me which **KMS key** I'll use to encrypt and decrypt the volume and I'll talk more about that in a second but by default now, because I've got it set on for the entire region, I don't need to explicitly enable it, so I'm just going to skip by these next steps. I'll create a new security group. I'll set the security group name and the description to encryption demo. Make sure that it's got SSH access enabled on the security group, and I'll change that to my IP address. I'll review and launch, click on Launch, create a new key pair, and I'll call this one encryption demo as well. Download the private component of that key pair and then launch the instance. Now while that instance is creating, I want to talk about the architecture.

So how does EBS encryption work? Well, it actually cooperates with a service called **KMS, which is the key management service and that's located within IAM**. **So the encryption and decryption the EBS performs is handled in part by KMS. KMS is a service that manages encryption keys and the primary type of encryption key is known as a CMK or customer master key.**

There are two types of customer master keys: **AWS managed keys and customer managed keys**. There as the name suggests with AWS managed keys they're provided and managed by AWS and you've got a very limited amount of control over those. Customer managed keys could be created by the customer, and you've got additional control around elements of that key, **including key rotation.**

Now by default, when any service within AWS wants to use encryption, it creates an AWS managed key inside the particular region. **So KMS is a regional service**. We've got North Virginia selected and if I go to AWS managed keys because I've already decided to use encryption within EBS it's created this encryption key. So anything that's AWS/ is an AWS managed customer master key.

Now when we selected the account default settings so if I go back to the dashboard and click on settings. There was also an option to pick a particular default encryption key now because we didn't select it, it's utilizing this default service key for EBS in this particular region. We could change this if we wanted to, but just for this demonstration there's no point doing so. Only if you've got particular key rotation requirements would you change.

**It's not actually KMS or customer master keys that encrypt the physical volumes.**

This is what happens: whenever encryption is required, **the service asks KMS to generate what's known as a data encryption key or DEK. Now KMS generates two versions of these data encryption keys, an encrypted version and a plain text version. The encrypted version of that data encryption key could be stored along with whatever is being encrypted, so it's perfectly safe to store it along with the encrypted volume within the EBS service. The decrypted version of that key is sent to the EC2 host that's running this instance, and it's only ever held in memory on that EC2 host while this instance is active. The minute this instance moves, the minute this host is rebooted, or the minute I shut down the instance, the plain text version of this key is discarded, and you can only get back that plain text version by requesting that KMS decrypts the encrypted version and sends it back.**

So there's always this secondary entity which is KMS controlling access to these plain text keys. **So that means in order to encrypt or decrypt disks, not only do you need the relevant IAM permissions to perform those activities, you also need to be able to interact with KMS. So KMS generates these customer master keys. Those never leave the device.** **EBS requests KMS to generate some data encryption keys, and then it delivers those keys back to EBS, an encrypted version that's stored along with the encrypted data and a plain text version that you used to actually encrypt and decrypt that data in transit.** **Now every single volume that you create inside EC2, as long as you're creating it blank from fresh uses its own dedicated data encryption key and that's generated by KMS.**

From the operating system's perspective, this is not an encrypted volume. It has no concept of the volume being encrypted. Now the reason for that is the encryption happens on the EC2 host, and there's no loss of performance whatsoever on the EC2 instance, as far as the EC2 instance is concerned, its running on an encrypted data. The data is essentially encrypted on the EC2 host, and it's transmitted in an encrypted form through to the EBS volumes where it's stored encrypted. When that data's needed, it's pulled off the EBS volumes into the EC2 host and these plain text data encryption keys are used to decrypt that data and present it to the operating system.

Every blank encrypted volume that you create inside EBS uses its own dedicated data encryption key. Now, **if you create a snapshot from a volume It's always going to be encrypted. You cannot create an unencrypted snapshot for an encrypted volume. You have to create an encrypted snapshot.** Also, remember that **this snapshot will be encrypted with the same data encryption key as the original volume and if you create any volumes from this snapshot, they also have to be encrypted, and they use the same data encryption key.**

If you have a copy of a snapshot to another region, you'll need to specify a destination encryption key to use. Remember, **KMS is a regional service. So by copying this snapshot to another region, you'll need to utilize another costumer master key, so you'll need to specify a new encryption key to use in the destination region.** Now remember early in the course, we also created an AMI and that uses snapshots. **So if you're creating AMIs from encrypted volumes than those AMIs or the snapshots in those AMIs will also be encrypted. So they have the same dependencies. They'll utilize the same encryption key structure, and if you move them or share them between accounts, you've got the same problem with dependencies on the keys.**

It's important to realize that whenever you use encryption, you've always got to think about the encryption keys that are used

EBS volume encryption. The key thing to remember for the exam is that it's handled by an EC2 host**. It doesn't cause any performance impact whatsoever on the instance**. In fact, the instance, as we've seen, is unaware that there is any encryption or decryption occurring. From its point of view, it is just an unencrypted file system. Now **EBS encryption means that the data is encrypted at rest and in transit to and from the EC2 host**.

One last thing, if you do get any exam questions that require you to manage the encryption keys and handle the encryption from an operating system perspective then EBS encryption is not going to be enough. For that you need to utilize a OS level encryption and most operating systems, including Linux and Microsoft Windows have got built-in capability to encrypt volumes. **So if you need to manage the key and if you need the operating system to manage the encryption process, then don't utilize EBS encryption utilize built-in operating system encryption.** You are able to use both but be aware from an exam question perspective **only operating system encryption will ensure that from an operating system perspective, the file's encrypted.** Remember, **from an operating system perspective, EBS encryption is transparent and you don't manage the keys, they're managed by KMS.** Now EBS encryption is supported by most instance types. There are some older generation instances which don't support it, but any of the current modern instance generations, especially those that use the nitro platform.