# Amazon EC2 Key Pairs

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a key pair.

To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance. Linux instances have no password, and you use a key pair to log in using SSH. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

## Creating a Key Pair Using Amazon EC2

You can create a key pair using the Amazon EC2 console or the command line. After you create a key pair, you can specify it when you launch your instance. You can also add the key pair to a running instance to enable another user to connect to the instance.

**To create your key pair using the Amazon EC2 console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.

**Tip**

The navigation pane is on the left side of the Amazon EC2 console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane.

1. Choose **Create Key Pair**.
2. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then choose **Create**.
3. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is .pem. Save the private key file in a safe place.

**Important**

This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

1. If you will use an SSH client on a Mac or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

**Copy**

$ **chmod 400 *my-key-pair*.pem**

**To import the public key**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Choose **Import Key Pair**.
4. In the **Import Key Pair** dialog box, choose **Browse**, and select the public key file that you saved previously. Enter a name for the key pair in the **Key pair name** field, and choose **Import**.

**To verify that your key pair was imported**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region in which you created the key pair.
3. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
4. Verify that the key pair that you imported is in the displayed list of key pairs.

**To delete your key pair using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.
3. Select the key pair and choose **Delete**.
4. When prompted, choose **Yes**.

## Connecting to Your Linux Instance if You Lose Your Private Key

If you lose the private key for an EBS-backed instance, you can regain access to your instance. You must stop the instance, detach its root volume and attach it to another instance as a data volume, modify the authorized\_keys file, move the volume back to the original instance, and restart the instance. This procedure isn't supported for instance store-backed instances. To determine the root device type of your instance, open the Amazon EC2 console, choose **Instances**, select the instance, and check the value of **Root device type** in the details pane. The value is either ebs or instance store. If the root device is an instance store volume, you must have the private key in order to connect to the instance.

**Prerequisites**

Create a new key pair using either the Amazon EC2 console or a third-party tool. If you want to name your new key pair exactly the same as the lost private key, you must first delete the existing key pair.

**To connect to an EBS-backed instance with a different key pair**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Instances** in the navigation pane, and then select the instance that you'd like to connect to. (We'll refer to this as the original instance.)
3. Save the following information that you'll need to complete this procedure.
   * Write down the instance ID, AMI ID, and Availability Zone of the original instance.
   * In the **Root device** field, take note of the device name for the root volume (for example, /dev/sda1 or /dev/xvda). Choose the link and write down the volume ID in the **EBS ID** field (vol-xxxxxxxxxxxxxxxxx).
   * [EC2-Classic] If the original instance has an associated Elastic IP address, write down the Elastic IP address shown in the **Elastic IP** field in the details pane.
4. Choose **Actions**, select **Instance State**, and then select **Stop**. If **Stop** is disabled, either the instance is already stopped or its root device is an instance store volume.

**Warning**

When you stop an instance, the data on any instance store volumes is erased. Therefore, if you have any data on instance store volumes that you want to keep, be sure to back it up to persistent storage.

1. Choose **Launch Instance**, and then use the launch wizard to launch a temporary instance with the following options:
   * On the **Choose an AMI** page, select the same AMI that you used to launch the original instance. If this AMI is unavailable, you can create an AMI that you can use from the stopped instance.
   * On the **Choose an Instance Type** page, leave the default instance type that the wizard selects for you.
   * On the **Configure Instance Details** page, specify the same Availability Zone as the instance you'd like to connect to. If you're launching an instance in a VPC, select a subnet in this Availability Zone.
   * On the **Add Tags** page, add the tag Name=Temporary to the instance to indicate that this is a temporary instance.
   * On the **Review** page, choose **Launch**. Create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.
2. In the navigation pane, choose **Volumes** and select the root device volume for the original instance (you wrote down its volume ID in a previous step). Choose **Actions**, and then select **Detach Volume**. Wait for the state of the volume to become available. (You might need to choose the **Refresh** icon.)
3. With the volume still selected, choose **Actions**, and then select **Attach Volume**. Select the instance ID of the temporary instance, write down the device name specified under **Device** (for example, /dev/sdf), and then choose **Yes, Attach**.

**Note**

If you launched your original instance from an AWS Marketplace AMI and your volume contains AWS Marketplace codes, you must first stop the temporary instance before you can attach the volume.

1. Connect to the temporary instance.
2. From the temporary instance, mount the volume that you attached to the instance so that you can access its file system. For example, if the device name is /dev/sdf, use the following commands to mount the volume as /mnt/tempvol.

**Note**

The device name may appear differently on your instance. For example, devices mounted as /dev/sdf may show up as /dev/xvdf on the instance. Some versions of Red Hat (or its variants, such as CentOS) may even increment the trailing letter by 4 characters, where /dev/sd*f*becomes /dev/xvd*k*.

* + Use the **lsblk** command to determine if the volume is partitioned.

**Copy**

[ec2-user ~]$ **lsblk**

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

xvda 202:0 0 8G 0 disk

└─xvda1 202:1 0 8G 0 part /

xvdf 202:80 0 101G 0 disk

└─xvdf1 202:81 0 101G 0 part

xvdg 202:96 0 30G 0 disk

In the above example, /dev/xvda and /dev/xvdf are partitioned volumes, and /dev/xvdg is not. If your volume is partitioned, you mount the partition (/dev/xvdf1) instead of the raw device (/dev/xvdf) in the next steps.

* + Create a temporary directory to mount the volume.

**Copy**

[ec2-user ~]$ **sudo mkdir /mnt/tempvol**

* + Mount the volume (or partition) at the temporary mount point, using the volume name or device name you identified earlier.

**Copy**

[ec2-user ~]$ **sudo mount /dev/*xvdf1* /mnt/tempvol**

1. From the temporary instance, use the following command to update authorized\_keys on the mounted volume with the new public key from the authorized\_keys for the temporary instance (you may need to substitute a different user name in the following command, such as ubuntu for Ubuntu instances):

**Copy**

[ec2-user ~]$ **cp .ssh/authorized\_keys /mnt/tempvol/home/*ec2-user*/.ssh/authorized\_keys**

If this copy succeeded, you can go to the next step.

(Optional) Otherwise, if you don't have permission to edit files in /mnt/tempvol, you'll need to update the file using **sudo** and then check the permissions on the file to verify that you'll be able to log into the original instance. Use the following command to check the permissions on the file:

**Copy**

[ec2-user ~]$ **sudo ls -l /mnt/tempvol/home/ec2-user/.ssh**

total 4

-rw------- 1 *222 500* 398 Sep 13 22:54 authorized\_keys

In this example output, *222* is the user ID and *500* is the group ID. Next, use **sudo** to re-run the copy command that failed:

**Copy**

[ec2-user ~]$ **sudo cp .ssh/authorized\_keys /mnt/tempvol/home/*ec2-user*/.ssh/authorized\_keys**

Run the following command again to determine whether the permissions changed:

**Copy**

[ec2-user ~]$ **sudo ls -l /mnt/tempvol/home/*ec2-user*/.ssh**

If the user ID and group ID have changed, use the following command to restore them:

**Copy**

[ec2-user ~]$ **sudo chown *222:500* /mnt/tempvol/home/*ec2-user*/.ssh/authorized\_keys**

1. From the temporary instance, unmount the volume that you attached so that you can reattach it to the original instance. For example, use the following command to unmount the volume at /mnt/tempvol:

**Copy**

[ec2-user ~]$ **sudo umount /mnt/tempvol**

1. From the Amazon EC2 console, select the volume with the volume ID that you wrote down, choose **Actions**, and then select **Detach Volume**. Wait for the state of the volume to become available. (You might need to choose the **Refresh** icon.)
2. With the volume still selected, choose **Actions**, **Attach Volume**. Select the instance ID of the original instance, specify the device name you noted earlier for the original root device attachment (/dev/sda1 or /dev/xvda), and then choose **Yes, Attach**.

**Warning**

If you don't specify the same device name as the original attachment, you cannot start the original instance. Amazon EC2 expects the root device volume at sda1 or /dev/xvda.

1. Select the original instance, choose **Actions**, select **Instance State**, and then choose **Start**. After the instance enters the running state, you can connect to it using the private key file for your new key pair.

**Note**

If the name of your new key pair and corresponding private key file is different to the name of the original key pair, ensure that you specify the name of the new private key file when you connect to your instance.

1. [EC2-Classic] If the original instance had an associated Elastic IP address before you stopped it, you must re-associate it with the instance as follows:
   * In the navigation pane, choose **Elastic IPs**.
   * Select the Elastic IP address that you wrote down at the beginning of this procedure.
   * Choose **Actions**, and then select **Associate address**.
   * Select the ID of the original instance, and then choose **Associate**.
2. (Optional) You can terminate the temporary instance if you have no further use for it. Select the temporary instance, choose **Actions**, select **Instance State**, and then choose **Terminate**.