Troubleshoot issues with Amazon EC2 instances

The following procedures and tips can help you troubleshoot issues with your Amazon EC2 instances.

Issues

- Troubleshoot Amazon EC2 instance launch issues
- Troubleshoot Amazon EC2 instance stop issues
- Troubleshoot Amazon EC2 instance termination issues
- Troubleshoot an unreachable Amazon EC2 instance
- Troubleshoot issues connecting to your Amazon EC2 Linux instance
- Troubleshoot Amazon EC2 Linux instances with failed status checks
- Troubleshoot an Amazon EC2 Linux instance booting from wrong volume
- Troubleshoot issues connecting to your Amazon EC2 Windows instance
- Troubleshoot Amazon EC2 Windows instance start issues
- Troubleshoot issues with Amazon EC2 Windows instances
- Reset the Windows administrator password for an Amazon EC2 Windows instance
- Troubleshoot Sysprep issues with Amazon EC2 Windows instances
- Troubleshoot impaired Amazon EC2 Linux instance using EC2Rescue
- Troubleshoot impaired Amazon EC2 Windows instance using EC2Rescue
- EC2 Serial Console for instances
- Send a diagnostic interrupt to debug an unreachable Amazon EC2 instance

Troubleshoot Amazon EC2 instance launch issues

The following are troubleshooting tips to help you solve issues when launching an Amazon EC2 instance.

Launch Issues

- Invalid device name
- Instance limit exceeded
- Insufficient instance capacity

- The requested configuration is currently not supported. Please check the documentation for supported configurations.
- Instance terminates immediately
- Insufficient permissions
- High CPU usage shortly after Windows starts (Windows instances only)

Invalid device name

Description

You get the `Invalid device name` *`device_name`* error when you try to launch a new instance.

Cause

If you get this error when you try to launch an instance, the device name specified for one or more volumes in the request has an invalid device name. Possible causes include:

- The device name might be in use by the selected AMI.
- The device name might be reserved for root volumes.
- The device name might be used for another volume in the request.
- The device name might not be valid for the operating system.

Solution

To resolve the issue:

- Ensure that the device name is not used in the AMI that you selected. Run the following command to view the device names used by the AMI.

```
aws ec2 describe-images --image-id ami-0abcdef1234567890 --query
 'Images[*].BlockDeviceMappings[].DeviceName'
```

- Ensure that you are not using a device name that is reserved for root volumes. For more information, see Available device names.
- Ensure that each volume specified in your request has a unique device name.
- Ensure that the device names that you specified are in the correct format. For more information, see Available device names.

Instance limit exceeded

Description

You get the `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get an `InstanceLimitExceeded` error when you try to launch a new instance or restart a stopped instance, you have reached the limit on the number of instances that you can launch in a Region. When you create your AWS account, we set default limits on the number of instances you can run on a per-Region basis.

Solution

You can request an instance limit increase on a per-region basis. For more information, see Amazon EC2 service quotas.

Insufficient instance capacity

Description

You get the `InsufficientInstanceCapacity` error when you try to launch a new instance or restart a stopped instance.

Cause

If you get this error when you try to launch an instance or restart a stopped instance, AWS does not currently have enough available On-Demand capacity to fulfill your request.

Solution

To resolve the issue, try the following:

- Wait a few minutes and then submit your request again; capacity can shift frequently.
- Submit a new request with a reduced number of instances. For example, if you're making a single request to launch 15 instances, try making 3 requests for 5 instances, or 15 requests for 1 instance instead.
- If you're launching an instance, submit a new request without specifying an Availability Zone.

- If you're launching an instance, submit a new request using a different instance type (which you can resize at a later stage). For more information, see Amazon EC2 instance type changes.
- If you are launching instances into a cluster placement group, you can get an insufficient capacity error.

The requested configuration is currently not supported. Please check the documentation for supported configurations.

Description

You get the `Unsupported` error when you try to launch a new instance because the instance configuration is not supported.

Cause

The error message provides additional details. For example, an instance type or instance purchasing option might not be supported in the specified Region or Availability Zone.

Solution

Try a different instance configuration. To search for an instance type that meets your requirements, see Find an Amazon EC2 instance type.

Instance terminates immediately

Description

Your instance goes from the `pending` state to the `terminated` state.

Cause

The following are a few reasons why an instance might immediately terminate:

- You've exceeded your EBS volume limits. For more information, see Amazon EBS volume limits for Amazon EC2 instances.
- An EBS snapshot is corrupted.
- The root EBS volume is encrypted and you do not have permissions to access the KMS key for decryption.

- A snapshot specified in the block device mapping for the AMI is encrypted and you do not have permissions to access the KMS key for decryption or you do not have access to the KMS key to encrypt the restored volumes.
- The Amazon S3-backed AMI that you used to launch the instance is missing a required part (an image.part.*xx* file).

For more information, get the termination reason using one of the following methods.

To get the termination reason using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and select the instance.
3. On the first tab, find the reason next to **State transition reason**.

To get the termination reason using the AWS CLI

1. Use the describe-instances command and specify the instance ID.

```
aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

2. Review the JSON response returned by the command and note the values in the `StateReason` response element.

   The following code block shows an example of a `StateReason` response element.

```
"StateReason": {
  "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",
  "Code": "Server.InternalError"
},
```

To get the termination reason using AWS CloudTrail

For more information, see Viewing events with CloudTrail event history in the *AWS CloudTrail User Guide*.

Solution

Depending on the termination reason, take one of the following actions:

- **`Client.VolumeLimitExceeded: Volume limit exceeded`** — Delete unused volumes. You can submit a request to increase your volume limit.
- **`Client.InternalError: Client error on launch`** — Ensure that you have the permissions required to access the AWS KMS keys used to decrypt and encrypt volumes. For more information, see Using key policies in AWS KMS in the *AWS Key Management Service Developer Guide*.

Insufficient permissions

Description

You get the `"errorMessage": "You are not authorized to perform this operation."` error when you try to launch a new instance, and the launch fails.

Cause

If you get this error when you try to launch an instance, you don't have the required IAM permissions to launch the instance.

Possible missing permissions include:

- `ec2:RunInstances`
- `iam:PassRole`

Other permissions might also be missing. For the list of permissions required to launch an instance, see the example IAM policies under Example: Use the EC2 launch instance wizard and Launch instances (RunInstances).

Solution

To resolve the issue:

- If you are making requests as an IAM user, verify that you have the following permissions:
  - `ec2:RunInstances` with a wildcard resource ("*")
  - `iam:PassRole` with the resource matching the role ARN (for example, `arn:aws:iam::999999999999:role/ExampleRoleName`)
- If you don't have the preceding permissions, edit the IAM policy associated with the IAM role or user to add the missing required permissions.

If your issue is not resolved and you continue receiving a launch failure error, you can decode the authorization failure message included in the error. The decoded message includes the permissions that are missing from the IAM policy. For more information, see How do I decode an authorization failure message after I receive an "UnauthorizedOperation" error during an EC2 instance launch?

High CPU usage shortly after Windows starts (Windows instances only)

> **Note**
>
> This troubleshooting tip is for Windows instances only.

If Windows Update is set to **Check for updates but let me choose whether to download and install them** (the default instance setting) this check can consume anywhere from 50 - 99% of the CPU on the instance. If this CPU consumption causes problems for your applications, you can manually change Windows Update settings in **Control Panel** or you can use the following script in the Amazon EC2 user data field:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update" /v
 AUOptions /t REG_DWORD /d 3 /f net stop wuauserv net start wuauserv
```

When you run this script, specify a value for /d. The default value is 3. Possible values include the following:

1. Never check for updates
2. Check for updates but let me choose whether to download and install them
3. Download updates but let me choose whether to install them
4. Install updates automatically

After you modify the user data for your instance, you can run it. For more information, see Run commands on your Windows instance at launch.

Troubleshoot Amazon EC2 instance stop issues

If your Amazon EBS-backed instance appears stuck in the `stopping` state, the issue might be with the underlying host computer.

To resolve the issue, follow these steps:

1. **Force stop the instance**

   Use the Amazon EC2 console or the AWS CLI to force stop the instance. For the steps, see Force stop an instance.

   The instance will first attempt a graceful shutdown, which includes flushing file system caches and metadata (although you can optionally bypass the graceful shutdown). If the graceful shutdown fails to complete within the timeout period, the instance shuts down forcibly without flushing the file system caches and metadata.

2. **After force stop**

   Perform file system check and repair procedures.

   > **Important**
   >
   > Performing these procedures is crucial because a forced stop prevents flushing of file system caches and metadata.

3. **If force stop fails**

   If, after 10 minutes, the instance has not stopped, do the following:

   a. Post a request for help on AWS re:Post. To help expedite a resolution, include the instance ID, and describe the steps that you've already taken.

   b. Alternatively, if you have a support plan, create a technical support case in the Support Center.

   c. While waiting for assistance, you can create a replacement instance if needed. For the steps, see (Optional) Create a replacement instance.

There is no cost for instance usage while an instance is in the `stopping` state or in any other state except `running`. You are only charged for instance usage when an instance is in the `running` state.

Contents

Force stop an instance

You can force an instance to stop. If, after 10 minutes, the instance has not stopped, post a request for help on AWS re:Post. To help expedite a resolution, include the instance ID, and describe the steps that you've already taken. Alternatively, if you have a support plan, create a technical support case in the Support Center.

> **Note**
>
> Using the console, you can force an instance to stop while the instance is in the `stopping` state only. Using the AWS CLI, you can force an instance to stop while the instance is in the `pending`, `running`, or `stopping` state.

Console

To force stop an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Instance state**, **Force stop instance**.

   Note that **Force stop instance** is only available in the console if your instance is in the `stopping` state. If your instance is in another state (except `shutting-down` and `terminated`) you can use the AWS CLI to force stop your instance.
4. (Optional) To bypass the graceful OS shutdown during the force stop, select the **Skip OS shutdown** checkbox.
5. Choose **Force stop**.

AWS CLI

To force stop an instance

Use the stop-instances command with the `--force` option.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

To bypass the graceful OS shutdown during force stop, include the `--skip-os-shutdown` option.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

To force stop an instance

Use the Stop-EC2Instance cmdlet and set `-Enforce` to `true`.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

To bypass the graceful OS shutdown during force stop, include `-SkipOsShutdown $true`.

```
Stop-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

(Optional) Create a replacement instance

While you are waiting for assistance from AWS re:Post or the Support Center, you can create a replacement instance if needed. Create an AMI from the stuck instance, and launch a new instance using the new AMI.

> **Important**
>
> You can create a replacement instance if the stuck instance produces system status checks only, as instance status checks will result in the AMI copying over an exact replica of the broken operating system. After you've confirmed the status message, create the AMI and launch a new instance using the new AMI.

Console

To create a replacement instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Actions**, **Image and templates**, **Create image**.
4. On the **Create image** page, do the following:

   a. Enter a name and description for the AMI.

   b. Clear **Reboot instance**.

   c. Choose **Create image**.

   For more information, see the section called “Create an AMI from an instance”.
5. Launch a new instance from the AMI and verify that the new instance is working.
6. Select the stuck instance, and choose **Actions**, **Instance state**, **Terminate (delete) instance**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

If you are unable to create an AMI from the instance as described in the previous procedure, you can set up a replacement instance as follows:

(Alternate) To create a replacement instance using the console

1. Select the instance and choose **Description**, **Block devices**. Select each volume and make note of its volume ID. Be sure to note which volume is the root volume.
2. In the navigation pane, choose **Volumes**. Select each volume for the instance, and choose **Actions**, **Create Snapshot**.
3. In the navigation pane, choose **Snapshots**. Select the snapshot that you just created, and choose **Actions**, **Create Volume**.
4. Launch an instance with the same operating system as the stuck instance. Note the volume ID and device name of its root volume.
5. In the navigation pane, choose **Instances**, select the instance that you just launched, and choose **Instance state**, **Stop instance**.

6. In the navigation pane, choose **Volumes**, select the root volume of the stopped instance, and choose **Actions**, **Detach Volume**.
7. Select the root volume that you created from the stuck instance, choose **Actions**, **Attach Volume**, and attach it to the new instance as its root volume (using the device name that you made note of). Attach any additional non-root volumes to the instance.
8. In the navigation pane, choose **Instances** and select the replacement instance. Choose **Instance state**, **Start instance**. Verify that the instance is working.
9. Select the stuck instance, choose **Instance state**, **Terminate (delete) instance**. If the instance also gets stuck terminating, Amazon EC2 automatically forces it to terminate within a few hours.

AWS CLI

To create a replacement instance

1. Create an AMI from the stuck instance using the create-image command with the `--no-reboot` option.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "my-replacement-ami" \
    --description "AMI for replacement instance" \
    --no-reboot
```

2. Launch a new instance from the AMI that you just created, using the run-instances command.
3. Verify that the new instance is working.
4. (Optional) Terminate the stuck instance using the terminate-instances command.

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

PowerShell

To create a replacement instance

1. Create an AMI from the stuck instance using the New-EC2Image cmdlet and set `-NoReboot` to `true`.

```
New-EC2Image `
    -InstanceId i-1234567890abcdef0 `
    -Name "my-replacement-ami" `
    -Description "AMI for replacement instance" `
    -NoReboot $true
```

2. Launch a new instance from the AMI that you just created, using the New-EC2Instance cmdlet.
3. Verify that the new instance is working.
4. (Optional) Terminate the stuck instance using the Remove-EC2Instance cmdlet.

```
Remove-EC2Instance -InstanceId i-1234567890abcdef0
```

Troubleshoot Amazon EC2 instance termination issues

Shutting down or deleting your instance is known as instance termination. The following information can help you troubleshoot issues when you terminate your instance.

You are not billed for any instance usage while an instance is not in the `running` state. In other words, when you terminate an instance, you stop incurring charges for that instance as soon as its state changes to `shutting-down`.

Instance terminates immediately

Several issues can cause your instance to terminate immediately on start-up. See Instance terminates immediately for more information.

Delayed instance termination

If your instance remains in the `shutting-down` state longer than a few minutes, it might be because:

- The instance is running shutdown scripts.
- There's a problem with the underlying host computer.

After several hours in the `shutting-down` state, Amazon EC2 treats the instance as stuck and forcibly terminates it.

To resolve a stuck instance yourself:

1. **Force terminate the instance**

   Use the Amazon EC2 console or the AWS CLI to force terminate the instance. For the steps, see Force terminate an instance.

   The instance will first attempt a graceful shutdown, which includes flushing file system caches and metadata (although you can optionally bypass the graceful shutdown). If the graceful shutdown fails to complete within the timeout period, the instance shuts down forcibly without flushing the file system caches and metadata.

2. **If force terminate fails**

   If, after several hours, the instance has not terminated and it appears stuck terminating, do the following:

   a. Post a request for help on AWS re:Post. To help expedite a resolution, include the instance ID, and describe the steps that you've already taken.

   b. Alternatively, if you have a support plan, create a technical support case in the Support Center.

Force terminate an instance

If it appears that your instance is stuck terminating, you can force your instance to terminate. If, after several hours, the instance has not terminated, post a request for help to AWS re:Post. To help expedite a resolution, include the instance ID and describe the steps that you've already taken. Alternatively, if you have a support plan, create a technical support case in the Support Center.

Console

To force terminate an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and select the stuck instance.
3. Choose **Instance state**, **Force terminate instance**.

   Note that **Force terminate instance** is only available in the console if your instance is in the `stopping` state. If your instance is in another state (except `shutting-down` and `terminated`) you can use the AWS CLI to force terminate your instance.

4. (Optional) To bypass the graceful OS shutdown during the force terminate, select the **Skip OS shutdown** checkbox.
5. Choose **Force terminate**.

AWS CLI

To force terminate an instance

Use the terminate-instances command with the `--force` option.

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force
```

To bypass the graceful OS shutdown during force terminate, include the `--skip-os-shutdown` option.

```
aws ec2 terminate-instances \
    --instance-ids i-1234567890abcdef0 \
    --force \
    --skip-os-shutdown
```

PowerShell

To force terminate an instance

Use the Remove-EC2Instance cmdlet and set `-Enforce` to `true`.

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true
```

To bypass the graceful OS shutdown during force terminate, include `-SkipOsShutdown $true`.

```
Remove-EC2Instance `
    -InstanceId i-1234567890abcdef0 `
    -Enforce $true `
    -SkipOsShutdown $true
```

Terminated instance still displayed

After you terminate an instance, it remains visible for a short while before being deleted. The state shows as `terminated`. If the entry is not deleted after several hours, contact Support.

Error: The instance may not be terminated. Modify its 'disableApiTermination' instance attribute

If you try to terminate an instance and get the `The instance i-1234567890abcdef0 may not be terminated. Modify its 'disableApiTermination' instance attribute` error message, it indicates that the instance has been enabled for termination protection. Termination protection prevents the instance from being accidentally terminated.

You must disable termination protection before you can terminate the instance.

For more information, see Change instance termination protection.

Instances automatically launched or terminated

Generally, the following behaviors mean that you've used Amazon EC2 Auto Scaling, EC2 Fleet, or Spot Fleet to scale your computing resources automatically based on criteria that you've defined:

- You terminate an instance and a new instance launches automatically.
- You launch an instance and one of your instances terminates automatically.
- You stop an instance and it terminates and a new instance launches automatically.

To stop automatic scaling, find the Auto Scaling group or the fleet that is launching the instances and either set its capacity to 0 or delete it.

Troubleshoot an unreachable Amazon EC2 instance

The following information can help you troubleshoot unreachable Amazon EC2 instances. You can capture screenshots or access console output to help diagnose problems and determine if you should reboot your instance. For unreachable Windows instances, troubleshoot by reviewing screenshots returned by the service.

Contents

- Instance console output
- Capture a screenshot of an unreachable instance
- Common screenshots to troubleshoot unreachable Windows instances
- Instance recovery when a host computer fails
- Instance appeared offline and unexpectedly rebooted

Instance reboot

The ability to reboot instances that are otherwise unreachable is valuable for both troubleshooting and general instance management.

Just as you can reset a computer by pressing the reset button, you can reset EC2 instances using the Amazon EC2 console, CLI, or API. For more information, see Reboot your Amazon EC2 instance.

Instance console output

Console output is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started.

- **Linux instances** – The instance console output displays the exact console output that would normally be displayed on a physical monitor attached to a computer. The console output returns buffered information that was posted shortly after an instance transition state (start, stop, reboot, and terminate). The posted output is not continuously updated; only when it is likely to be of the most value.
- **Windows instances** – The instance console output includes the last three system event log errors.

Only the instance owner can access the console output.

You can retrieve the latest serial console output during the instance lifecycle. This option is only supported on Nitro-based instances.

Console

To get console output

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2. In the left navigation pane, choose **Instances**.
3. Select the instance.
4. Choose **Actions**, **Monitor and troubleshoot**, **Get system log**.

AWS CLI

To get console output

Use the get-console-output command.

```
aws ec2 get-console-output --instance-id i-1234567890abcdef0
```

PowerShell

To get console output

Use the Get-EC2ConsoleOutput cmdlet.

```
Get-EC2ConsoleOutput -InstanceId i-1234567890abcdef0
```

Capture a screenshot of an unreachable instance

If you are unable to connect to your instance, you can capture a screenshot of your instance and view it as an image. The image can provide visibility as to the status of the instance, and allows for quicker troubleshooting.

You can generate screenshots while the instance is running or after it has crashed. The image is generated in JPG format and is no larger than 100 kb. There is no data transfer cost for the screenshot.

Limitations

This feature is not supported for the following:

- Bare metal instances (instances of type `*.metal`)
- Instance is using an NVIDIA GRID driver
- Instances powered by Arm-based Graviton processors

- Windows instances on AWS Outposts
- Windows instances on AWS Local Zones

Region support

This feature is not available in the following Regions:

- Asia Pacific (Thailand)
- Mexico (Central)
- GovCloud Regions

Console

To get a screenshot of an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the left navigation pane, choose **Instances**.
3. Select the instance to capture.
4. Choose **Actions**, **Monitor and troubleshoot**, **Get instance screenshot**.
5. Choose **Download**, or right-click the image to download and save it.

AWS CLI

To capture a screenshot of an instance

Use the get-console-screenshot command. The output is base64-encoded.

```
aws ec2 get-console-screenshot --instance-id i-1234567890abcdef0
```

PowerShell

To capture a screenshot of an instance

Use the Get-EC2ConsoleScreenshot cmdlet. The output is base64-encoded.

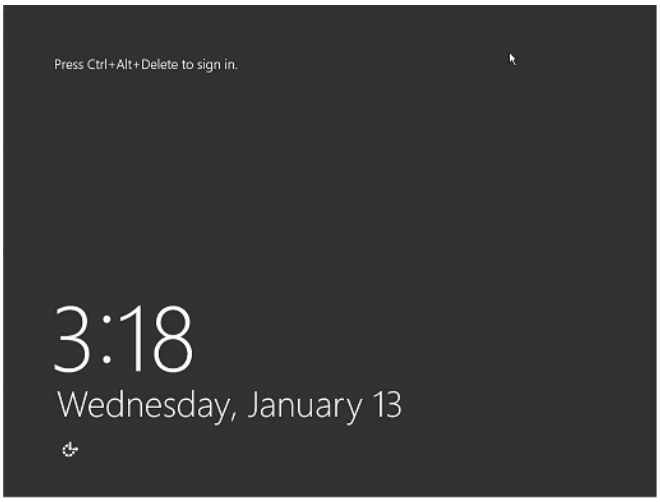
```
Get-EC2ConsoleScreenshot -InstanceId i-1234567890abcdef0
```

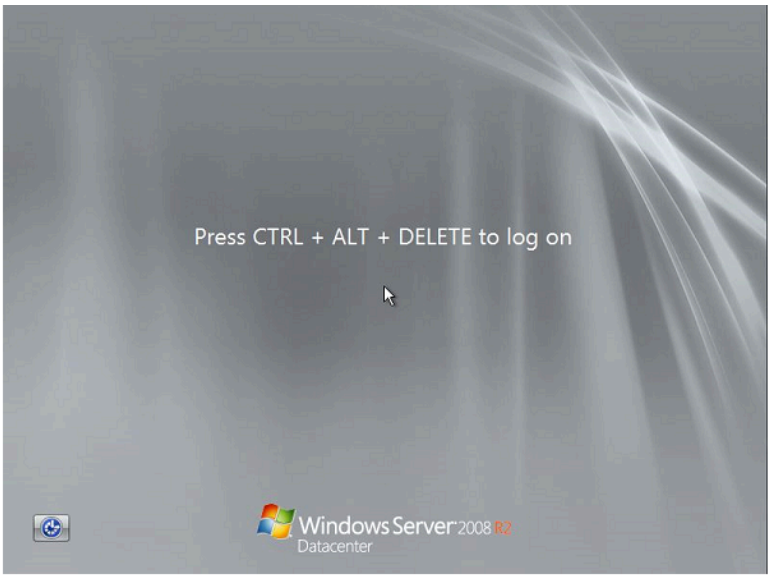
Common screenshots to troubleshoot unreachable Windows instances

You can use the following information to help you troubleshoot an unreachable Windows instance based on screenshots returned by the service.

Log on screen (Ctrl+Alt+Delete)

Console Screenshot Service returned the following.





If an instance becomes unreachable during logon, there could be a problem with your network configuration or Windows Remote Desktop Services. An instance can also be unresponsive if a process is using large amounts of CPU.

Network configuration

Use the following information to verify that your AWS, Microsoft Windows, and local (or on-premises) network configurations aren't blocking access to the instance.

AWS network configuration

Configuration	Verify
Security group configuration	Verify that port 3389 is open for your security group. Verify you are connecting to the right public IP address. If the instance was not associated with an Elastic IP, the public IP changes after the instance stops/starts. For more information, see Remote Desktop can't connect to the remote computer.
VPC configuration (Network ACLs)	Verify that the access control list (ACL) for your Amazon VPC is not blocking access. For information, see Network ACLs in the *Amazon VPC User Guide*.
VPN configuration	If you are connecting to your VPC using a virtual private network (VPN), verify VPN tunnel connectivity. For more information, see Troubleshooting AWS Client VPN: Tunnel connectivity issues to a VPC.

Windows network configuration

Configuration	Verify
Windows Firewall	Verify that Windows Firewall isn't blocking connections to your instance. Disable Windows Firewall as described in bullet 7 of the remote desktop troubleshooting section, Remote Desktop can't connect to the remote computer.
Advanced TCP/IP configuration (Use of static IP)	The instance may be unresponsive because you configured a static IP address. For a VPC, create a network interface and attach it to the instance.

Local or on-premises network configuration

Verify that a local network configuration isn't blocking access. Try to connect to another instance in the same VPC as your unreachable instance. If you can't access another instance, work with your local network administrator to determine whether a local policy is restricting access.

Remote Desktop Services issues

If the instance can't be reached during logon, there could a problem with Remote Desktop Services (RDS) on the instance.

> **Tip**
>
> You can use the `AWSSupport-TroubleshootRDP` runbook to check and modify various settings that might affect Remote Desktop Protocol (RDP) connections. For more information, see `AWSSupport-TroubleshootRDP` in the *AWS Systems Manager Automation runbook reference*.

Remote Desktop Services configuration

Configuration	Verify
RDS is running	Verify that RDS is running on the instance. Connect to the instance using the Microsoft Management Console (MMC) Services snap-in (`services.msc`). In the list of services, verify that **Remote Desktop Services** is **Running**. If it isn't, start it and then set the startup type to **Automatic**. If you can't connect to the instance by using the Services snap-in, detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same Availability Zone as a secondary volume, and modify the Start registry key. When you are finished, reattach the root volume to the original instance.
RDS is enabled	Even if the service is started, it might be disabled. Detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, attach the original volume to another instance in the same Availability Zone as a secondary volume,

Configuration	Verify
	and enable the service by modifying the **Terminal Server** registry key as described in Enable Remote Desktop on an EC2 instance with remote registry. When you are finished, reattach the root volume to the original instance.

High CPU usage

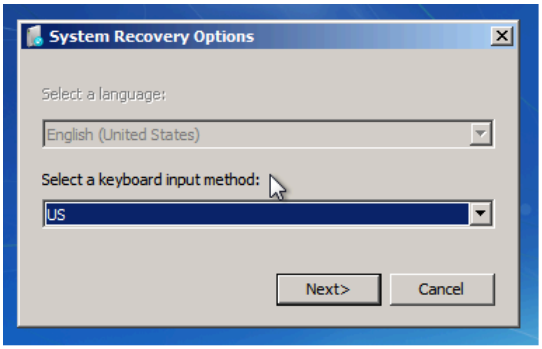
Check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch. If **CPUUtilization (Maximum)** is a high number, wait for the CPU to go down and try connecting again. High CPU usage can be caused by:

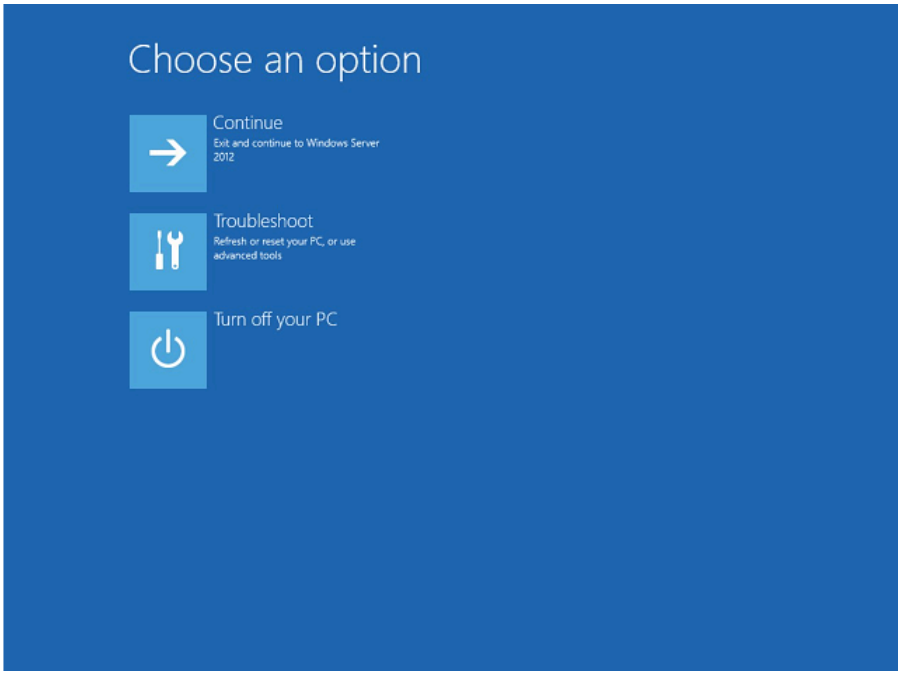
- Windows Update
- Security Software Scan
- Custom Startup Script
- Task Scheduler

For more information, see Get Statistics for a Specific Resource in the *Amazon CloudWatch User Guide*. For additional troubleshooting tips, see High CPU usage shortly after Windows starts (Windows instances only).

Recovery console screen

Console Screenshot Service returned the following.





The operating system might boot into the Recovery console and get stuck in this state if the `bootstatuspolicy` is not set to `ignoreallfailures`. Use the following procedure to change the `bootstatuspolicy` configuration to `ignoreallfailures`.

By default, the policy configuration for public Windows AMIs provided by AWS is set to `ignoreallfailures`.

1. Stop the unreachable instance.
2. Create a snapshot of the root volume. The root volume is attached to the instance as `/dev/sda1`.

   Detach the root volume from the unreachable instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same Availability Zone as a secondary volume.

   > **Warning**
   >
   > If your temporary instance and the original instance were launched using the same AMI, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. If you must create a temporary instance using the same AMI, to avoid a disk signature collision, complete the steps in Disk signature collision.

Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses an AMI for Windows Server 2016, launch the temporary instance using an AMI for Windows Server 2019.

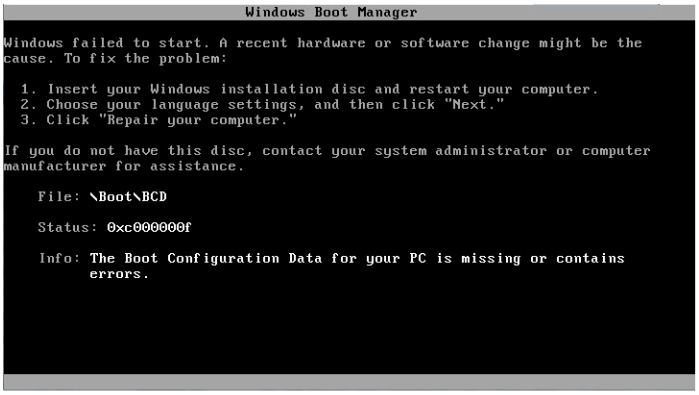
3. Log in to the instance and run the following command from a command prompt to change the `bootstatuspolicy` configuration to `ignoreallfailures`.

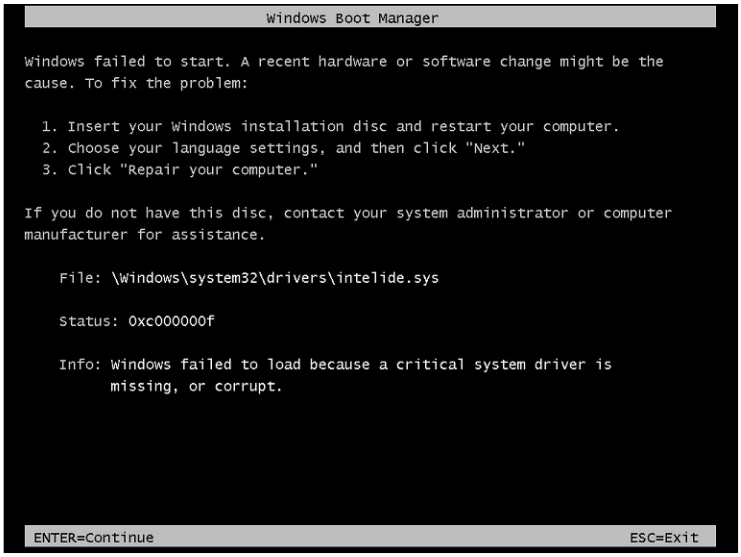
```
bcdedit /store Drive Letter:\boot\bcd /set {default} bootstatuspolicy
 ignoreallfailures
```

4. Reattach the volume to the unreachable instance and start the instance again.

Windows boot manager screen

Console Screenshot Service returned the following.

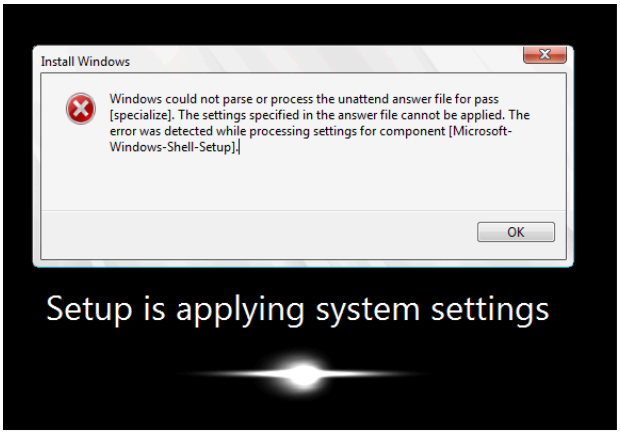


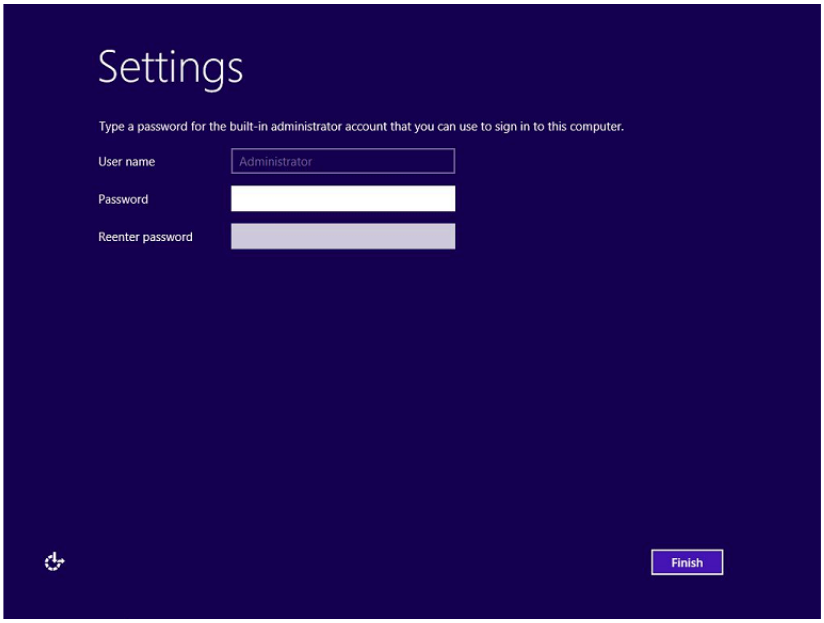


The operating system experienced a fatal corruption in the system file and/or the registry. When the instance is stuck in this state, you should recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach any root volumes from the unreachable instance, take a snapshot of those volume or create an AMI from them, and attach them to another instance in the same Availability Zone as a secondary volume.

Sysprep screen

Console Screenshot Service returned the following.

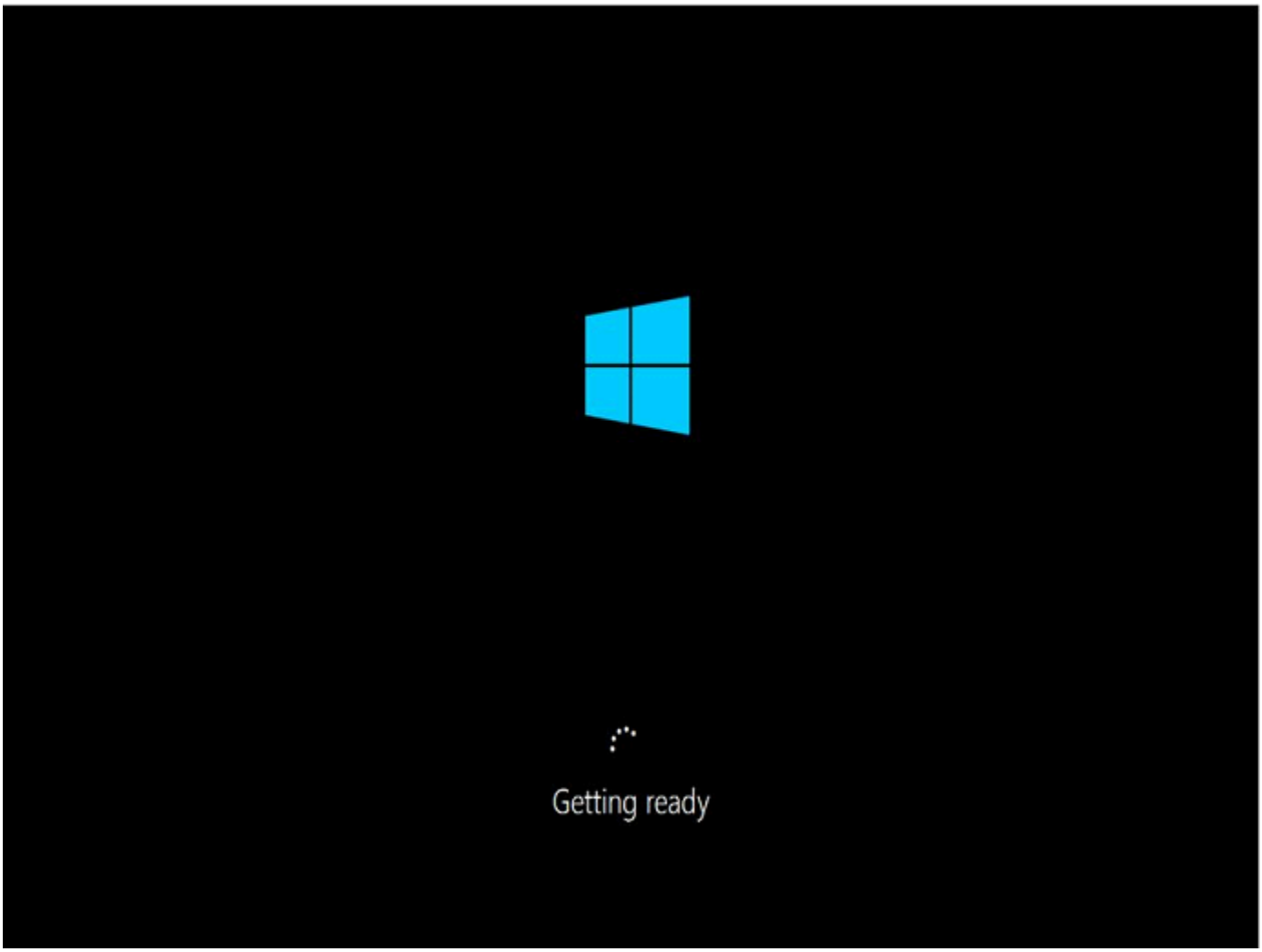




You may see this screen if you did not use the EC2Config Service to call Sysprep or if the operating system failed while running Sysprep. You can reset the password using EC2Rescue. Otherwise, see Create an Amazon EC2 AMI using Windows Sysprep.

Getting ready screen

Console Screenshot Service returned the following.



Refresh the Instance Console Screenshot Service repeatedly to verify that the progress ring is spinning. If the ring is spinning, wait for the operating system to start up. You can also check the **CPUUtilization (Maximum)** metric on your instance by using Amazon CloudWatch to see if the operating system is active. If the progress ring is not spinning, the instance may be stuck at the boot process. Reboot the instance. If rebooting does not solve the problem, recover the instance from a recent backup AMI or launch a replacement instance. If you need to access data on the instance, detach the root volume from the unreachable instance, take a snapshot of the volume or create an AMI from it. Then attach it to another instance in the same Availability Zone as a secondary volume.

Windows Update screen

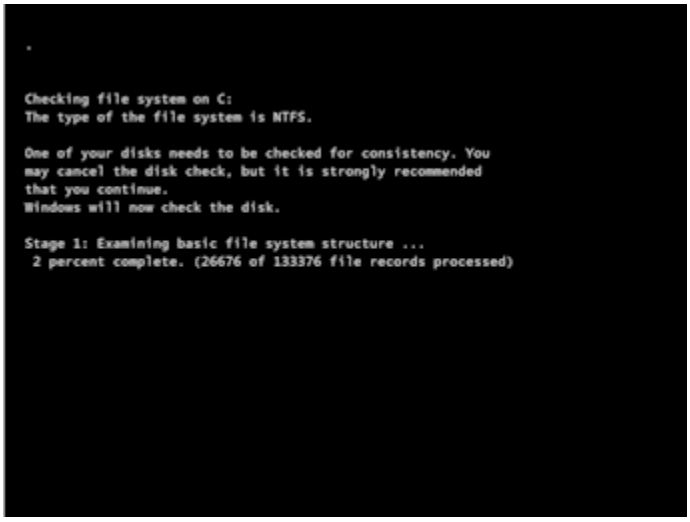
Console Screenshot Service returned the following.

The Windows Update process is updating the registry. Wait for the update to finish. Do not reboot or stop the instance as this may cause data corruption during the update.

> **Note**
>
> The Windows Update process can consume resources on the server during the update. If you experience this problem often, consider using faster instance types and faster EBS volumes.

Chkdsk

Console Screenshot Service returned the following.



Windows is running the chkdsk system tool on the drive to verify file system integrity and fix logical file system errors. Wait for process to complete.

Instance recovery when a host computer fails

If there is an unrecoverable issue with the hardware of an underlying host computer, AWS may schedule an instance stop event. You are notified of such an event ahead of time by email.

To recover an Amazon EBS-backed instance running on a host computer that failed

1. Back up any important data on your instance store volumes to Amazon EBS or Amazon S3.
2. Stop the instance.
3. Start the instance.
4. Restore any important data.

For more information, see Stop and start Amazon EC2 instances.

To recover an instance with an instance store root volume running on a host computer that failed

1. Create an AMI from the instance.
2. Upload the image to Amazon S3.
3. Back up important data to Amazon EBS or Amazon S3.
4. Terminate the instance.
5. Launch a new instance from the AMI.
6. Restore any important data to the new instance.

Instance appeared offline and unexpectedly rebooted

If your instance appears to have been offline and then unexpectedly rebooted, it might have undergone automatic instance recovery. This happens when AWS detects that the instance is unavailable due to an underlying hardware or software issue, and either simplified automatic recovery or CloudWatch action based recovery is enabled on the instance.

During the recovery process, AWS attempts to restore the instance's availability by migrating it to different hardware. To verify whether automatic instance recovery occurred for your instance, see Verify if automatic instance recovery occurred.

Note

If your workload or application is unresponsive, check whether it's running on the instance. If it's not, start it manually. To prevent this issue in the future, implement a recovery plan to ensure your workload or application functions properly after instance recovery.

Troubleshoot issues connecting to your Amazon EC2 Linux instance

The following information and common errors can help you troubleshoot connecting to your Linux instance.

Connection issues

- Common causes for connection issues
- Error connecting to your instance: Connection timed out
- Error: unable to load key ... Expecting: ANY PRIVATE KEY
- Error: User key not recognized by server
- Error: Permission denied or connection closed by [instance] port 22
- Error: Unprotected private key file
- Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"
- Error: Host key verification failed
- Error: Server refused our key or No supported authentication methods available
- Cannot ping instance
- Error: Server unexpectedly closed network connection
- Error: Host key validation failed for EC2 Instance Connect
- Can't connect to Ubuntu instance using EC2 Instance Connect
- I've lost my private key. How can I connect to my instance?

Common causes for connection issues

We recommend that you start to troubleshoot instance connection problems by verifying that you have accurately performed the following tasks.

Verify the username for your instance

You can connect to your instance using the username for your user account or the default username for the AMI that you used to launch your instance.

- **Get the username for your user account.**

  For more information about how to create a user account, see Manage system users on your Amazon EC2 Linux instance.

- **Get the default username for the AMI that you used to launch your instance.**

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	`admin`
Fedora	`fedora` or `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
Other	Check with the AMI provider

Verify that your security group rules allow traffic

Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. The default security group for the VPC does not allow incoming SSH traffic by default. The security group created by the launch instance wizard enables SSH traffic by default. For steps to add a rule for inbound SSH traffic to your Linux instance, see Rules to connect to instances from your computer. For steps to verify, see Error connecting to your instance: Connection timed out.

Verify that your instance is ready

After you launch an instance, it can take a few minutes for the instance to be ready to accept connection requests. Check your instance to make sure it is running and has passed its status checks.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. Verify the following:
   a. In the **Instance state** column, verify that your instance is in the `running` state.
   b. In the **Status check** column, verify that your instance has passed all status checks.

Verify that you've satisfied all prerequisites to connect

Ensure that you have all the information that you need to connect. For more information, see Connect to your Linux instance using SSH.

Connect from Linux or macOS X

If your local computer operating system is Linux or macOS X, check the following for specific prerequisites for connecting to a Linux instance:

- SSH client
- EC2 Instance Connect
- AWS Systems Manager Session Manager

Connect from Windows

If your local computer operating system is Windows, check the following for specific prerequisites for connecting to a Linux instance:

- OpenSSH
- PuTTY
- AWS Systems Manager Session Manager
- Windows Subsystem for Linux

Check if the instance is a managed instance

User-initiated connections to managed instances are not allowed. To determine if the instance is managed, find the **Managed** field for the instance. If the value is **true**, it's a managed instance. For more information, see Amazon EC2 managed instances.

Error connecting to your instance: Connection timed out

If you try to connect to your instance and get the error message `Network error: Connection timed out` or `Error connecting to [instance], reason: -> Connection timed out: connect`, try the following:

Check your security group rules.

You need a security group rule that allows inbound traffic from your local computer's public IPv4 address on the proper port.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Security** tab at the bottom of the console page, under **Inbound rules**, check the list of rules that are in effect for the selected instance. Verify that there is a rule that allows traffic from your local computer to port 22 (SSH).

   If your security group does not have a rule that allows inbound traffic from your local computer, add a rule to your security group. For more information, see Rules to connect to instances from your computer.
4. For the rule that allows inbound traffic, check the **Source** field. If the value is a single IP address, and if the IP address is not static, a new IP address will be assigned each time you restart your computer. This will result in the rule not including your computer's IP address traffic. The IP address might not be static if your computer is on a corporate network, or you're connecting through an internet service provider (ISP), or your computer IP address is dynamic and changes each time you restart your computer. To ensure that your security group rule

allows inbound traffic from your local computer, instead of specifying a single IP address for **Source**, rather specify the range of IP addresses used by your client computers.

For more information about security group rules, see Security group rules in the *Amazon VPC User Guide*.

Check the route table for the subnet.

You need a route that sends all traffic destined outside the VPC to the internet gateway for the VPC.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. On the **Networking** tab, make note of the values for **VPC ID** and **Subnet ID**.
4. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
5. In the navigation pane, choose **Internet Gateways**. Verify that there is an internet gateway attached to your VPC. Otherwise, choose **Create internet gateway**, enter a name for the internet gateway, and choose **Create internet gateway**. Then, for the internet gateway you created, choose **Actions**, **Attach to VPC**, select your VPC, and then choose **Attach internet gateway** to attach it to your VPC.
6. In the navigation pane, choose **Subnets**, and then select your subnet.
7. On the **Route table** tab, verify that there is a route with `0.0.0.0/0` as the destination and the internet gateway for your VPC as the target. If you're connecting to your instance using its IPv6 address, verify that there is a route for all IPv6 traffic (`::/0`) that points to the internet gateway. Otherwise, do the following:

   a. Choose the ID of the route table (rtb-*xxxxxxxx*) to navigate to the route table.

   b. On the **Routes** tab, choose **Edit routes**. Choose **Add route**, use `0.0.0.0/0` as the destination and the internet gateway as the target. For IPv6, choose **Add route**, use `::/0` as the destination and the internet gateway as the target.

   c. Choose **Save routes**.

Check the network access control list (ACL) for the subnet.

The network ACLs must allow inbound SSH traffic from your local IP address on port 22. It must also allow outbound traffic to the ephemeral ports (1024-65535).

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Subnets**.
3. Select your subnet.
4. On the **Network ACL** tab, for **Inbound rules**, verify that the rules allow inbound traffic from your computer on the required port. Otherwise, delete or modify the rule that is blocking the traffic.
5. For **Outbound rules**, verify that the rules allow outbound traffic to your computer on the ephemeral ports. Otherwise, delete or modify the rule that is blocking the traffic.

If your computer is on a corporate network

Ask your network administrator whether the internal firewall allows inbound and outbound traffic from your computer on port 22.

If you have a firewall on your computer, verify that it allows inbound and outbound traffic from your computer on port 22.

Check that your instance has a public IPv4 address.

If not, you can associate an Elastic IP address with your instance. For more information, see Elastic IP addresses.

Check the CPU load on your instance; the server may be overloaded.

AWS automatically provides data such as Amazon CloudWatch metrics and instance status, which you can use to see how much CPU load is on your instance and, if necessary, adjust how your loads are handled. For more information, see Monitor your instances using CloudWatch.

- If your load is variable, you can automatically scale your instances up or down using Auto Scaling and Elastic Load Balancing.
- If your load is steadily growing, you can move to a larger instance type. For more information, see Amazon EC2 instance type changes.

To connect to your instance using an IPv6 address, check the following:

- Your subnet must be associated with a route table that has a route for IPv6 traffic (`::/0`) to an internet gateway.

- Your security group rules must allow inbound traffic from your local IPv6 address on port 22.
- Your network ACL rules must allow inbound and outbound IPv6 traffic.
- If you launched your instance from an older AMI, it might not be configured for DHCPv6 (IPv6 addresses are not automatically recognized on the network interface). For more information, see Configure IPv6 on your instances in the *Amazon VPC User Guide*.
- Your local computer must have an IPv6 address, and must be configured to use IPv6.

Error: unable to load key ... Expecting: ANY PRIVATE KEY

If you try to connect to your instance and get the error message, `unable to load key ... Expecting: ANY PRIVATE KEY`, the file in which the private key is stored is incorrectly configured. If the private key file ends in `.pem`, it might still be incorrectly configured. A possible cause for an incorrectly configured private key file is a missing certificate.

If the private key file is incorrectly configured, follow these steps to resolve the error

1. Create a new key pair. For more information, see Create a key pair using Amazon EC2.

   > **Note**
   >
   > Alternatively, you can create a new key pair using a third-party tool. For more information, see Create a key pair using a third-party tool and import the public key to Amazon EC2.

2. Add the new key pair to your instance. For more information, see I've lost my private key. How can I connect to my instance?.
3. Connect to your instance using the new key pair.

Error: User key not recognized by server

If you use SSH to connect to your instance

- Use `ssh -vvv` to get triple verbose debugging information while connecting:

```
ssh -vvv -i path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

The following sample output demonstrates what you might see if you were trying to connect to your instance with a key that was not recognized by the server:

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-
interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

If you use PuTTY to connect to your instance

- Verify that your private key (.pem) file has been converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see Connect to your Linux instance using PuTTY.

> **Note**
>
> In PuTTYgen, load your private key file and select **Save Private Key** rather than **Generate**.

- Verify that you are connecting with the appropriate username for your AMI. Enter the username in the **Host name** box in the **PuTTY Configuration** window.

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	`admin`
Fedora	`fedora` or `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
Other	Check with the AMI provider

- Verify that you have an inbound security group rule to allow inbound traffic to the appropriate port. For more information, see Rules to connect to instances from your computer.

Error: Permission denied or connection closed by [instance] port 22

If you connect to your instance using SSH and get any of the following errors, `Host key not found in [directory]`, `Permission denied (publickey)`, `Authentication failed, permission denied`, or `Connection closed by [instance] port 22`, verify that you are connecting with the appropriate username for your AMI *and* that you have specified the proper private key (`.pem`) file for your instance.

The appropriate usernames are as follows:

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`
Debian	`admin`
Fedora	`fedora` or `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
Other	Check with the AMI provider

For example, to use an SSH client to connect to an Amazon Linux instance, use the following command:

```
ssh -i /path/key-pair-name.pem instance-user-
name@ec2-203-0-113-25.compute-1.amazonaws.com
```

Confirm that you are using the private key file that corresponds to the key pair that you selected when you launched the instance.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then select your instance.
3. On the **Details** tab, under **Instance details**, verify the value of **Key pair name**.
4. If you did not specify a key pair when you launched the instance, you can terminate the instance and launch a new instance, ensuring that you specify a key pair. If this is an instance that you have been using but you no longer have the `.pem` file for your key pair, you can replace the key pair with a new one. For more information, see I've lost my private key. How can I connect to my instance?.

If you generated your own key pair, ensure that your key generator is set up to create RSA keys. DSA keys are not accepted.

If you get a `Permission denied (publickey)` error and none of the above applies (for example, you were able to connect previously), the permissions on the home directory of your instance may have been changed. Permissions for /home/*instance-user-name*/.ssh/authorized_keys must be limited to the owner only.

To verify the permissions on your instance

1. Stop your instance and detach the root volume. For more information, see Stop and start Amazon EC2 instances.
2. Launch a temporary instance in the same Availability Zone as your current instance (use a similar or the same AMI as you used for your current instance), and attach the root volume to the temporary instance.
3. Connect to the temporary instance, create a mount point, and mount the volume that you attached.
4. From the temporary instance, check the permissions of the /home/*instance-user-name*/ directory of the attached volume. If necessary, adjust the permissions as follows:

```
[ec2-user ~]$ chmod 600 mount_point/home/instance-user-name/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/instance-user-name
```

5. Unmount the volume, detach it from the temporary instance, and re-attach it to the original instance. Ensure that you specify the correct device name for the root volume; for example, `/dev/xvda`.
6. Start your instance. If you no longer require the temporary instance, you can terminate it.

Error: Unprotected private key file

Your private key file must be protected from read and write operations from any other users. If your private key can be read or written to by anyone but you, then SSH ignores your key and you see the following warning message below.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0777 for '.ssh/my_private_key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: .ssh/my_private_key.pem
Permission denied (publickey).
```

If you see a similar message when you try to log in to your instance, examine the first line of the error message to verify that you are using the correct public key for your instance. The above example uses the private key `.ssh/my_private_key.pem` with file permissions of `0777`, which allow anyone to read or write to this file. This permission level is very insecure, and so SSH ignores this key.

If you are connecting from macOS or Linux, run the following command to fix this error, substituting the path for your private key file.

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

If you are connecting to a Linux instance from Windows, perform the following steps on your local computer.

1. Navigate to your .pem file.
2. Right-click on the .pem file and select **Properties**.
3. Choose the **Security** tab.
4. Select **Advanced**.
5. Verify that you are the owner of the file. If not, change the owner to your username.
6. Select **Disable inheritance** and **Remove all inherited permissions from this object**.
7. Select **Add**, **Select a principal**, enter your username, and select **OK**.
8. From the **Permission Entry** window, grant **Read** permissions and select **OK**.
9. Click **Apply** to ensure all settings are saved.
10. Select **OK** to close the **Advanced Security Settings** window.
11. Select **OK** to close the **Properties** window.
12. You should be able to connect to your Linux instance from Windows using SSH.

From a Windows command prompt, run the following commands.

1. From the command prompt, navigate to the file path location of your .pem file.
2. Run the following command to reset and remove explicit permissions:

```
icacls.exe $path /reset
```

3. Run the following command to grant Read permissions to the current user:

```
icacls.exe $path /GRANT:R "$($env:USERNAME):(R)"
```

4. Run the following command to disable inheritance and remove inherited permissions.

```
icacls.exe $path /inheritance:r
```

5. You should be able to connect to your Linux instance from Windows using SSH.

Error: Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"

If you use a third-party tool, such as **ssh-keygen**, to create an RSA key pair, it generates the private key in the OpenSSH key format. When you connect to your instance, if you use the private key in

the OpenSSH format to decrypt the password, you'll get the error `Private key must begin with "-----BEGIN RSA PRIVATE KEY-----" and end with "-----END RSA PRIVATE KEY-----"`.

To resolve the error, the private key must be in the PEM format. Use the following command to create the private key in the PEM format:

```
ssh-keygen -m PEM
```

Error: Host key verification failed

This error occurs if there is a mismatch between the host key stored on the instance in the `known_hosts` file and on the client. For example, a mismatch can occur if you connect to an instance using one public IP address, and then try to connect to it again using a different public IP address. This can happen after you add or remove an Elastic IP address, as doing so changes the public IP address of an instance.

To resolve this error, start by confirming that there was an expected change to the host key or the network configuration of the instance. Before you connect to the instance, you might also want to verify the host fingerprint. After you connect to the instance, you can remove the old host key from the `known_hosts` file. For instructions, refer to the documentation for the Linux distribution in use on your instance.

Error: Server refused our key *or* No supported authentication methods available

If you use PuTTY to connect to your instance and get either of the following errors, Error: Server refused our key or Error: No supported authentication methods available, verify that you are connecting with the appropriate username for your AMI. Type the username in **User name** in the **PuTTY Configuration** window.

The appropriate usernames are as follows:

AMI used to launch instance	Default username
Amazon Linux	`ec2-user`
CentOS	`centos` or `ec2-user`

AMI used to launch instance	Default username
Debian	`admin`
Fedora	`fedora` or `ec2-user`
FreeBSD	`ec2-user`
RHEL	`ec2-user` or `root`
SUSE	`ec2-user` or `root`
Ubuntu	`ubuntu`
Oracle	`ec2-user`
Bitnami	`bitnami`
Rocky Linux	`rocky`
Other	Check with the AMI provider

You should also verify that:

- You are using the latest version of PuTTY. For more information, see the PuTTY web page.
- Your private key (.pem) file has been correctly converted to the format recognized by PuTTY (.ppk). For more information about converting your private key, see Connect to your Linux instance using PuTTY.

Cannot ping instance

The `ping` command is a type of ICMP traffic — if you are unable to ping your instance, ensure that your inbound security group rules allow ICMP traffic for the `Echo Request` message from all sources, or from the computer or instance from which you are issuing the command.

If you are unable to issue a `ping` command from your instance, ensure that your outbound security group rules allow ICMP traffic for the `Echo Request` message to all destinations, or to the host that you are attempting to ping.

`Ping` commands can also be blocked by a firewall or time out due to network latency or hardware issues. You should consult your local network or system administrator for help with further troubleshooting.

Error: Server unexpectedly closed network connection

If you are connecting to your instance with PuTTY and you receive the error "Server unexpectedly closed network connection," verify that you have enabled keepalives on the Connection page of the PuTTY Configuration to avoid being disconnected. Some servers disconnect clients when they do not receive any data within a specified period of time. Set the Seconds between keepalives to 59 seconds.

If you still experience issues after enabling keepalives, try to disable Nagle's algorithm on the Connection page of the PuTTY Configuration.

Error: Host key validation failed for EC2 Instance Connect

If you rotate your instance host keys, the new host keys are not automatically uploaded to the AWS trusted host keys database. This causes host key validation to fail when you try to connect to your instance using the EC2 Instance Connect browser-based client, and you're unable to connect to your instance.

To resolve the error, you must run the `eic_harvest_hostkeys` script on your instance, which uploads your new host key to EC2 Instance Connect. The script is located at `/opt/aws/bin/` on Amazon Linux 2 instances, and at `/usr/share/ec2-instance-connect/` on Ubuntu instances.

Amazon Linux 2

To resolve the host key validation failed error on an Amazon Linux 2 instance

1. Connect to your instance using SSH.

   You can connect by using the EC2 Instance Connect CLI or by using the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For Amazon Linux 2, the default username is `ec2-user`.

   For example, if your instance was launched using Amazon Linux 2, your instance's public DNS name is `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, and the key pair is `my_ec2_private_key.pem`, use the following command to SSH into your instance:

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see Connect to your Linux instance using an SSH client.

2. Navigate to the following folder.

```
[ec2-user ~]$ cd /opt/aws/bin/
```

3. Run the following command on your instance.

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

Note that a successful call results in no output.

You can now use the EC2 Instance Connect browser-based client to connect to your instance.

Ubuntu

To resolve the host key validation failed error on an Ubuntu instance

1. Connect to your instance using SSH.

You can connect by using the EC2 Instance Connect CLI or by using the SSH key pair that was assigned to your instance when you launched it and the default username of the AMI that you used to launch your instance. For Ubuntu, the default username is `ubuntu`.

For example, if your instance was launched using Ubuntu, your instance's public DNS name is `ec2-a-b-c-d.us-west-2.compute.amazonaws.com`, and the key pair is `my_ec2_private_key.pem`, use the following command to SSH into your instance:

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-
west-2.compute.amazonaws.com
```

For more information about connecting to your instance, see Connect to your Linux instance using an SSH client.

2. Navigate to the following folder.

```
[ec2-user ~]$ cd /usr/share/ec2-instance-connect/
```

3. Run the following command on your instance.

```
[ec2-user ~]$ ./eic_harvest_hostkeys
```

Note that a successful call results in no output.

You can now use the EC2 Instance Connect browser-based client to connect to your instance.

Can't connect to Ubuntu instance using EC2 Instance Connect

If you use EC2 Instance Connect to connect to your Ubuntu instance and you get an error when attempting to connect, you can use the following information to try to fix the issue.

Possible cause

The `ec2-instance-connect` package on the instance is not the latest version.

Solution

Update the `ec2-instance-connect` package on the instance to the latest version, as follows:

1. Connect to your instance using a method other than EC2 Instance Connect.
2. Run the following command on your instance to update the `ec2-instance-connect` package to the latest version.

```
apt update && apt upgrade
```

I've lost my private key. How can I connect to my instance?

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 with a new public key, move the volume back to the original instance, and restart the instance. For more information about launching, connecting to, and stopping instances, see Amazon EC2 instance state changes.

This procedure is only supported for instances with EBS root volumes. If the instance has an instance store root volume, you cannot use this procedure to regain access to your instance; you must have the private key to connect to the instance. To determine the root volume type of your instance, open the Amazon EC2 console, choose **Instances**, select the instance, choose the **Storage** tab, and in the **Root device details** section, check the value of **Root device type**.

The value is either `EBS` or `INSTANCE-STORE`.

In addition to the following steps, there are other ways to connect to your Linux instance if you lose your private key. For more information, see How can I connect to my Amazon EC2 instance if I lost my SSH key pair after its initial launch?

Steps for connecting to an EBS-backed instance with a different key pair

- Step 1: Create a new key pair
- Step 2: Get information about the original instance and its root volume
- Step 3: Stop the original instance
- Step 4: Launch a temporary instance
- Step 5: Detach the root volume from the original instance and attach it to the temporary instance
- Step 6: Add the new public key to authorized_keys on the original volume mounted to the temporary instance
- Step 7: Unmount and detach the original volume from the temporary instance, and reattach it to the original instance
- Step 8: Connect to the original instance using the new key pair
- Step 9: Clean up

Step 1: Create a new key pair

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. For information about creating a new key pair, see Create a key pair using Amazon EC2 or Create a key pair using a third-party tool and import the public key to Amazon EC2.

Step 2: Get information about the original instance and its root volume

Make note of the following information because you'll need it to complete this procedure.

To get information about your original instance

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. On the **Details** tab, make note of the instance ID and AMI ID.
4. On the **Networking** tab, make note of the Availability Zone.
5. On the **Storage** tab, under **Root device name**, make note of the device name for the root volume (for example, `/dev/xvda`). Then, under **Block devices**, find this device name and make note of the volume ID (for example, vol-0a1234b5678c910de).

Step 3: Stop the original instance

Choose **Instance state**, **Stop instance**. If this option is disabled, either the instance is already stopped or its root volume is an instance store volume.

> **Warning**
>
> When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

Step 4: Launch a temporary instance

To launch a temporary instance

1. In the navigation pane, choose **Instances**, and then choose **Launch instances**.
2. In the **Name and tags** section, for **Name**, enter **Temporary**.
3. In the **Application and OS Images** section, 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. For more information, see Create an Amazon EBS-backed AMI.
4. In the **Instance type** section, keep the default instance type.
5. In the **Key pair** section, for **Key pair name**, select the existing key pair to use or create a new one.
6. In the **Network settings** section, choose **Edit**, and then for **Subnet**, select a subnet in the same Availability Zone as the original instance.

7. In the **Summary** panel, choose **Launch**.

Step 5: Detach the root volume from the original instance and attach it to the temporary instance

1. In the navigation pane, choose **Volumes** and select the root volume for the original instance (you made note of its volume ID in a previous step). Choose **Actions**, **Detach volume**, and then choose **Detach**. 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**, and then choose **Attach volume**. Select the instance ID of the temporary instance, make note of the device name specified under **Device name** (for example, `/dev/sdf`), and then choose **Attach volume**.

> **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.

Step 6: Add the new public key to `authorized_keys` on the original volume mounted to the temporary instance

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

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

```
[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 preceding 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.

b. Create a temporary directory to mount the volume.

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

c. Mount the volume (or partition) at the temporary mount point, using the volume name or device name that you identified earlier. The required command depends on your operating system's file system. Note that the device name might appear differently on your instance. See the note in Step 6 for more information.

- Amazon Linux, Ubuntu, and Debian

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

- Amazon Linux 2, CentOS, SUSE Linux 12, and RHEL 7.x

```
[ec2-user ~]$ sudo mount -o nouuid /dev/xvdf1 /mnt/tempvol
```

> **Note**
>
> If you get an error stating that the file system is corrupt, run the following command to use the **fsck** utility to check the file system and repair any issues:
>
> ```
> [ec2-user ~]$ sudo fsck /dev/xvdf1
> ```

3. 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.

> **Important**
>
> The following examples use the Amazon Linux username `ec2-user`. You might need to substitute a different username, such as `ubuntu` for Ubuntu instances.

```
[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 must update the file using **sudo** and then check the permissions on the file to verify that you are able to log into the original instance. Use the following command to check the permissions on the file.

```
[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.

```
[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.

```
[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.

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

Step 7: Unmount and detach the original volume from the temporary instance, and reattach it to the original instance

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`.

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

2. Detach the volume from the temporary instance (you unmounted it in the previous step): From the Amazon EC2 console, choose **Volumes** in the navigation pane, select the root volume for the original instance (you made note of the volume ID in a previous step), choose **Actions**, **Detach volume**, and then choose **Detach**. Wait for the state of the volume to become `available`. (You might need to choose the Refresh icon.)
3. Reattach the volume to the original instance: With the volume still selected, choose **Actions**, **Attach volume**. Select the instance ID of the original instance, specify the device name that you noted earlier in Step 2 for the original root volume attachment (`/dev/sda1` or `/dev/xvda`), and then choose **Attach volume**.

   Important

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

Step 8: Connect to the original instance using the new key pair

Select the original instance, choose **Instance state**, **Start instance**. 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 from 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.

Step 9: Clean up

(Optional) You can terminate the temporary instance if you have no further use for it. Select the temporary instance, and choose **Instance state**, **Terminate (delete) instance**.

Troubleshoot Amazon EC2 Linux instances with failed status checks

The following information can help you troubleshoot issues if your Linux instance fails a status check. First determine whether your applications are exhibiting any problems. If you verify that the instance is not running your applications as expected, review the status check information and the system logs.

For examples of problems that can cause status checks to fail, see Status checks for Amazon EC2 instances.

Contents

- Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)
- XENBUS: Device with no driver...
- ... days without being checked, check forced (File system check required)
- fsck died with exit status... (Missing device)
- GRUB prompt (grubdom>)
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)
- XENBUS: Timeout connecting to devices (Xenbus timeout)

Review status check information

To investigate impaired instances using the Amazon EC2 console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and then select your instance.
3. Select the **Status and alarms** tab to see the individual results for all **System status checks**, **Instance status checks**, and **Attached EBS status checks**.

If a status check has failed, you can try one of the following options:

- Create an alarm to recover the instance in response to the failed status check. For more information, see Create alarms that stop, terminate, reboot, or recover an instance.
- (Instance status checks) If you changed the instance type to a Nitro-based instance, status checks fail if you migrated from an instance that does not have the required ENA and NVMe drivers. For more information, see Compatibility for changing the instance type.
- For an instance with an EBS root volume, stop and restart the instance. For more information, see Stop and start Amazon EC2 instances.
- For an instance with an instance store root volume, terminate the instance and launch a replacement instance. For more information, see Terminate Amazon EC2 instances.
- Wait for Amazon EC2 to resolve the issue.

- Contact Support or post your issue to AWS re:Post.
- If your instance is in an Auto Scaling group:
  - (System status checks and instance status checks) By default, Amazon EC2 Auto Scaling automatically launches a replacement instance. For more information, see Health checks for instances in an Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.
  - (Attached EBS status checks) You must configure Amazon EC2 Auto Scaling to automatically launch a replacement instance. For more information, see Monitor and replace Auto Scaling instances with impaired Amazon EBS volumes in the *Amazon EC2 Auto Scaling User Guide*.
- Retrieve the system log and look for errors. For more information, see Retrieve the system logs.

Retrieve the system logs

If an instance status check fails, you can reboot the instance and retrieve the system logs. The logs may reveal an error that can help you troubleshoot the issue. Rebooting clears unnecessary information from the logs.

To reboot an instance and retrieve the system log

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**, and select your instance.
3. Choose **Instance state**, **Reboot instance**. It might take a few minutes for your instance to reboot.
4. Verify that the problem still exists; in some cases, rebooting may resolve the problem.
5. When the instance is in the `running` state, choose **Actions**, **Monitor and troubleshoot**, **Get system log**.
6. Review the log that appears on the screen, and use the list of known system log error statements below to troubleshoot your issue.
7. If your issue is not resolved, you can post your issue to AWS re:Post.

Troubleshoot system log errors for Linux instances

For Linux instances that have failed an instance status check, such as the instance reachability check, verify that you followed the steps above to retrieve the system log. The following list contains some common system log errors and suggested actions you can take to resolve the issue for each error.

Memory Errors

- Out of memory: kill process
- ERROR: mmu_update failed (Memory management update failed)

Device Errors

- I/O error (block device failure)
- I/O ERROR: neither local nor remote disk (Broken distributed block device)

Kernel Errors

- request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)
- "FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)
- "FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)
- ERROR Invalid kernel (EC2 incompatible kernel)

File System Errors

- fsck: No such file or directory while trying to open... (File system not found)
- General error mounting filesystems (failed mount)
- VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)
- Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)
- XENBUS: Device with no driver...
- ... days without being checked, check forced (File system check required)
- fsck died with exit status... (Missing device)

Operating System Errors

- GRUB prompt (grubdom>)

- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)
- Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)
- XENBUS: Timeout connecting to devices (Xenbus timeout)

Out of memory: kill process

An out-of-memory error is indicated by a system log entry similar to the one shown below.

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

Potential cause

Exhausted memory

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Stop the instance, and modify the instance to use a different instance type, and start the instance again. For example, a larger or a memory-optimized instance type. • Reboot the instance to return it to a non-impaired status. The problem will probably occur again unless you change the instance type.
Instance store-backed	Do one of the following: • Terminate the instance and launch a new instance, specifying a different instance

For this instance type	Do this
	type. For example, a larger or a memory-optimized instance type. • Reboot the instance to return it to an unimpaired status. The problem will probably occur again unless you change the instance type.

ERROR: mmu_update failed (Memory management update failed)

Memory management update failures are indicated by a system log entry similar to the following:

```
...
Press `ESC' to enter the menu... 0   [H[J  Booting 'Amazon Linux 2011.09
 (2.6.35.14-95.38.amzn1.i686)'


root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=

en_US.UTF-8 KEYTABLE=us

initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img

ERROR: mmu_update failed with rc=-22
```

Potential cause

Issue with Amazon Linux

Suggested action

Post your issue to AWS re:Post or contact Support.

I/O error (block device failure)

An input/output error is indicated by a system log entry similar to the following example:

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance.

For this instance type	Do this
	2. Detach the volume. 3. Attempt to recover the volume. **Note** It's good practice to snapshot your Amazon EBS volumes often. This dramatically decreases the risk of data loss as a result of failure. 4. Re-attach the volume to the instance. 5. Start the instance.
Instance store-backed	Terminate the instance and launch a new instance. **Note** Data cannot be recovered. Recover from backups. **Note** It's a good practice to use either Amazon S3 or Amazon EBS for backups. Instance store volumes are directly tied to single host and single disk failures.

I/O ERROR: neither local nor remote disk (Broken distributed block device)

An input/output error on the device is indicated by a system log entry similar to the following example:

```
...
block drbd1: Local IO failed in request_timer_fn. Detaching...

Aborting journal on device drbd1-8.

block drbd1: IO ERROR: neither local nor remote disk

Buffer I/O error on device drbd1, logical block 557056

lost page write due to I/O error on drbd1

JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	A failed Amazon EBS volume
Instance store-backed	A failed physical drive

Suggested action

Terminate the instance and launch a new instance.

For an Amazon EBS-backed instance you can recover data from a recent snapshot by creating an image from it. Any data added after the snapshot cannot be recovered.

request_module: runaway loop modprobe (Looping legacy kernel modprobe on older Linux versions)

This condition is indicated by a system log similar to the one shown below. Using an unstable or old Linux kernel (for example, 2.6.16-xenU) can cause an interminable loop condition at startup.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

 Xen: 0000000000000000 - 0000000026700000 (usable)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c

request_module: runaway loop modprobe binfmt-464c
```

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use a newer kernel, either GRUB-based or static, using one of the following options: Option 1: Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` parameters. Option 2: 1. Stop the instance. 2. Modify the kernel and ramdisk attributes to use a newer kernel. 3. Start the instance.

For this instance type	Do this
Instance store-backed	Terminate the instance and launch a new instance, specifying the `-kernel` and `-ramdisk` parameters.

"FATAL: kernel too old" and "fsck: No such file or directory while trying to open /dev" (Kernel and AMI mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

Potential causes

Incompatible kernel and userland

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Modify the configuration to use a newer kernel. 3. Start the instance.
Instance store-backed	Use the following procedure: 1. Create an AMI that uses a newer kernel. 2. Terminate the instance.

For this instance type	Do this
	3. Start a new instance from the AMI you created.

"FATAL: Could not load /lib/modules" or "BusyBox" (Missing kernel modules)

This condition is indicated by a system log similar to the one shown below.

```
[    0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file or
 directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing: No
 such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device.  Common problems:
 - Boot args (cat /proc/cmdline)
   - Check rootdelay= (did the system wait long enough?)
   - Check root= (did the system wait for the right device?)
 - Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or
 directory
```

```
ALERT! /dev/sda1 does not exist. Dropping to a shell!


BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

Potential causes

One or more of the following conditions can cause this problem:

- Missing ramdisk
- Missing correct modules from ramdisk
- Amazon EBS root volume not correctly attached as `/dev/sda1`

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Select corrected ramdisk for the Amazon EBS volume. 2. Stop the instance. 3. Detach the volume and repair it. 4. Attach the volume to the instance. 5. Start the instance. 6. Modify the AMI to use the corrected ramdisk.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance with the correct ramdisk. 2. Create a new AMI with the correct ramdisk.

ERROR Invalid kernel (EC2 incompatible kernel)

This condition is indicated by a system log similar to the one shown below.

```
...
root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

  Booting 'Fallback'

root (hd0)

 Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

Potential causes

One or both of the following conditions can cause this problem:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure:

For this instance type	Do this
	1. Stop the instance. 2. Replace with working kernel. 3. Install a fallback kernel. 4. Modify the AMI by correcting the kernel.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance with the correct kernel. 2. Create an AMI with the correct kernel. 3. (Optional) Seek technical assistance for data recovery using Support.

fsck: No such file or directory while trying to open... (File system not found)

This condition is indicated by a system log similar to the one shown below.

```
  Welcome to Fedora
  Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [  OK  ]

Starting udev: [  OK  ]

Setting hostname localhost:  [  OK  ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[  OK  ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh
```

```
/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]


*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

Potential causes

- A bug exists in ramdisk filesystem definitions /etc/fstab
- Misconfigured filesystem definitions in /etc/fstab
- Missing/failed drive

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance, detach the root volume, repair/modify /etc/fstab the volume, attach the volume to the instance, and start the instance. 2. Fix ramdisk to include modified /etc/fstab (if applicable). 3. Modify the AMI to use a newer ramdisk.

For this instance type	Do this
	The sixth field in the fstab defines availability requirements of the mount – a nonzero value implies that an fsck will be done on that volume and *must* succeed. Using this field can be problematic in Amazon EC2 because a failure typically results in an interactive console prompt that is not currently available in Amazon EC2. Use care with this feature and read the Linux man page for fstab.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance. 2. Detach any errant Amazon EBS volumes and the reboot instance. 3. (Optional) Seek technical assistance for data recovery using Support.

General error mounting filesystems (failed mount)

This condition is indicated by a system log similar to the one shown below.

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting.  Commit interval 5 seconds
```

```
EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	• Detached or failed Amazon EBS volume. • Corrupted filesystem. • Mismatched ramdisk and AMI combination (such as Debian ramdisk with a SUSE AMI).
Instance store-backed	• A failed drive. • A corrupted file system. • A mismatched ramdisk and combination (for example, a Debian ramdisk with a SUSE AMI).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the root volume. 3. Attach the root volume to a known working instance. 4. Run filesystem check (fsck -a /dev/...). 5. Fix any errors. 6. Detach the volume from the known working instance. 7. Attach the volume to the stopped instance. 8. Start the instance. 9. Recheck the instance status.
Instance store-backed	Try one of the following: • Start a new instance. • (Optional) Seek technical assistance for data recovery using Support.

VFS: Unable to mount root fs on unknown-block (Root filesystem mismatch)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
 20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line:  root=/dev/sda1 ro 4
...
Registering block device major 8
...
```

```
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

Potential causes

Instance type	Potential cause
Amazon EBS-backed	• Device not attached correctly. • Root device not attached at correct device point. • Filesystem not in expected format. • Use of legacy kernel (such as 2.6.16-XenU). • A recent kernel update on your instance (faulty update, or an update bug)
Instance store-backed	Hardware device failure.

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Stop and then restart the instance. • Modify root volume to attach at the correct device point, possible /dev/sda1 instead of /dev/sda. • Stop and modify to use modern kernel. • Refer to the documentation for your Linux distribution to check for known update bugs. Change or reinstall the kernel.
Instance store-backed	Terminate the instance and launch a new instance using a modern kernel.

Error: Unable to determine major/minor number of root device... (Root file system/device mismatch)

This condition is indicated by a system log similar to the one shown below.

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda or sda instead of sda1)
- Incorrect choice of instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the volume. 3. Fix the device mapping problem.

For this instance type	Do this
	4. Start the instance. 5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: 1. Create a new AMI with the appropriate fix (map block device correctly). 2. Terminate the instance and launch a new instance from the AMI you created.

XENBUS: Device with no driver...

This condition is indicated by a system log similar to the one shown below.

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

Potential causes

- Missing or incorrectly configured virtual block device driver
- Device enumeration clash (sda versus xvda)
- Incorrect choice of instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the instance. 2. Detach the volume. 3. Fix the device mapping problem. 4. Start the instance. 5. Modify the AMI to address device mapping issues.
Instance store-backed	Use the following procedure: 1. Create an AMI with the appropriate fix (map block device correctly). 2. Terminate the instance and launch a new instance using the AMI you created.

... days without being checked, check forced (File system check required)

This condition is indicated by a system log similar to the one shown below.

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

Potential causes

Filesystem check time passed; a filesystem check is being forced.

Suggested actions

- Wait until the filesystem check completes. A filesystem check can take a long time depending on the size of the root filesystem.
- Modify your filesystems to remove the filesystem check (fsck) enforcement using tune2fs or tools appropriate for your filesystem.

fsck died with exit status... (Missing device)

This condition is indicated by a system log similar to the one shown below.

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

Potential causes

- Ramdisk looking for missing drive
- Filesystem consistency check forced
- Drive failed or detached

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Try one or more of the following to resolve the issue: • Stop the instance, attach the volume to an existing running instance. • Manually run consistency checks. • Fix ramdisk to include relevant utilities.

For this instance type	Do this
	• Modify filesystem tuning parameters to remove consistency requirements (not recommended).
Instance store-backed	Try one or more of the following to resolve the issue: • Rebundle ramdisk with correct tooling. • Modify file system tuning parameters to remove consistency requirements (not recommended). • Terminate the instance and launch a new instance. • (Optional) Seek technical assistance for data recovery using Support.

GRUB prompt (grubdom>)

This condition is indicated by a system log similar to the one shown below.

```
    GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

       [ Minimal BASH-like line editing is supported.   For

         the   first   word,  TAB  lists  possible  command

         completions.  Anywhere else TAB lists the possible

         completions of a device/filename. ]

grubdom>
```

Potential causes

Instance type	Potential causes
Amazon EBS-backed	• Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location. • Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).
Instance store-backed	• Missing GRUB configuration file. • Incorrect GRUB image used, expecting GRUB configuration file at a different location. • Unsupported filesystem used to store your GRUB configuration file (for example, converting your root file system to a type that is not supported by an earlier version of GRUB).

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Option 1: Modify the AMI and relaunch the instance: 1. Modify the source AMI to create a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary.

For this instance type	Do this
	3. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 4. Terminate the instance and launch a new one using the AMI that you created. Option 2: Fix the existing instance: 1. Stop the instance. 2. Detach the root filesystem. 3. Attach the root filesystem to a known working instance. 4. Mount filesystem. 5. Create a GRUB configuration file. 6. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 7. Detach filesystem. 8. Attach to the original instance. 9. Modify kernel attribute to use the appropria te GRUB image (1st disk or 1st partition on 1st disk). 10Start the instance.

For this instance type	Do this
Instance store-backed	Option 1: Modify the AMI and relaunch the instance: 1. Create the new AMI with a GRUB configuration file at the standard location (/boot/grub/menu.lst). 2. Pick the appropriate GRUB image, (hd0-1st drive or hd00 – 1st drive, 1st partition). 3. Verify that your version of GRUB supports the underlying file system type and upgrade GRUB if necessary. 4. Terminate the instance and launch a new instance using the AMI you created. Option 2: Terminate the instance and launch a new instance, specifying the correct kernel. **Note** To recover data from the existing instance, contact Support.

Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (Hard-coded MAC address)

This condition is indicated by a system log similar to the one shown below.

```
...
Bringing up loopback interface:  [  OK  ]

Bringing up interface eth0:  Device eth0 has different MAC address than expected,
 ignoring.
[FAILED]
```

```
Starting auditd: [  OK  ]
```

Potential causes

There is a hardcoded interface MAC in the AMI configuration

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Modify the AMI to remove the hardcoding and relaunch the instance. • Modify the instance to remove the hardcoded MAC address. OR Use the following procedure: 1. Stop the instance. 2. Detach the root volume. 3. Attach the volume to another instance and modify the volume to remove the hardcoded MAC address. 4. Attach the volume to the original instance. 5. Start the instance.
Instance store-backed	Do one of the following: • Modify the instance to remove the hardcoded MAC address. • Terminate the instance and launch a new instance.

Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. (SELinux misconfiguration)

This condition is indicated by a system log similar to the one shown below.

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

Potential causes

SELinux has been enabled in error:

- Supplied kernel is not supported by GRUB
- Fallback kernel does not exist

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Use the following procedure: 1. Stop the failed instance. 2. Detach the failed instance's root volume. 3. Attach the root volume to another running Linux instance (later referred to as a recovery instance). 4. Connect to the recovery instance and mount the failed instance's root volume. 5. Disable SELinux on the mounted root volume. This process varies across Linux distributions; for more information, consult your OS-specific documentation.

For this instance type	Do this
	Note On some systems, you disable SELinux by setting `SELINUX=disabled` in the `/mount_point /etc/sysconfig/selinux` file, where *mount_point* is the location that you mounted the volume on your recovery instance. 6. Unmount and detach the root volume from the recovery instance and reattach it to the original instance. 7. Start the instance.
Instance store-backed	Use the following procedure: 1. Terminate the instance and launch a new instance. 2. (Optional) Seek technical assistance for data recovery using Support.

XENBUS: Timeout connecting to devices (Xenbus timeout)

This condition is indicated by a system log similar to the one shown below.

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found.  Try passing init= option to kernel.
```

Potential causes

- The block device is not connected to the instance
- This instance is using an old instance kernel

Suggested actions

For this instance type	Do this
Amazon EBS-backed	Do one of the following: • Modify the AMI and instance to use a modern kernel and relaunch the instance. • Reboot the instance.
Instance store-backed	Do one of the following: • Terminate the instance. • Modify the AMI to use a modern kernel, and launch a new instance using this AMI.

Troubleshoot an Amazon EC2 Linux instance booting from wrong volume

In some situations, a volume other than the volume attached to `/dev/xvda` or `/dev/sda` becomes the root volume of a Linux instance. This can happen when you have attached the root volume of another instance, or a volume created from the snapshot of a root volume, to an instance with an existing root volume.

This is due to how the initial ramdisk in Linux works. It chooses the volume defined as `/` in the `/etc/fstab`, and in some distributions, this is determined by the label attached to the volume partition. Specifically, you find that your `/etc/fstab` looks something like the following:

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
```

```
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

If you check the label of both volumes, you see that they both contain the / label:

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

In this example, you could end up having /dev/xvdf1 become the root volume that your instance boots to after the initial ramdisk runs, instead of the /dev/xvda1 volume from which you had intended to boot. To solve this, use the same **e2label** command to change the label of the attached volume that you do not want to boot from.

In some cases, specifying a UUID in /etc/fstab can resolve this. However, if both volumes come from the same snapshot, or the secondary is created from a snapshot of the primary volume, they share a UUID.

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
 PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

To change the label of an attached ext4 volume

1. Use the **e2label** command to change the label of the volume to something other than /.

   ```
   [ec2-user ~]$ sudo e2label /dev/xvdf1 old/
   ```

2. Verify that the volume has the new label.

   ```
   [ec2-user ~]$ sudo e2label /dev/xvdf1
   old/
   ```

To change the label of an attached xfs volume

- Use the **xfs_admin** command to change the label of the volume to something other than /.

```
[ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1
writing all SBs
new label = "old/"
```

After changing the volume label as shown, you should be able to reboot the instance and have the proper volume selected by the initial ramdisk when the instance boots.

> **Important**
>
> If you intend to detach the volume with the new label and return it to another instance to use as the root volume, you must perform the above procedure again and change the volume label back to its original value. Otherwise, the other instance does not boot because the ramdisk is unable to find the volume with the label `/`.

Troubleshoot issues connecting to your Amazon EC2 Windows instance

The following information and common errors can help you troubleshoot issues when connecting to your Windows instance.

Connection issues

- Remote Desktop can't connect to the remote computer
- Error using the macOS RDP client
- RDP displays a black screen instead of the desktop
- Unable to remotely log on to an instance with a user that is not an administrator
- Troubleshooting Remote Desktop issues using AWS Systems Manager
- Enable Remote Desktop on an EC2 instance with remote registry
- I've lost my private key. How can I connect to my Windows instance?

Remote Desktop can't connect to the remote computer

Try the following to resolve issues related to connecting to your instance:

- Verify that you're using the correct public DNS hostname. (In the Amazon EC2 console, select the instance and check **Public DNS (IPv4)** in the details pane.) If your instance is in a VPC and you do not see a public DNS name, you must enable DNS hostnames. For more information, see DNS attributes for your VPC in the *Amazon VPC User Guide*.
- Verify that your instance has a public IPv4 address. If not, you can associate an Elastic IP address with your instance. For more information, see Elastic IP addresses.
- To connect to your instance using an IPv6 address, check that your local computer has an IPv6 address and is configured to use IPv6. For more information, see Configure IPv6 on your instances in the *Amazon VPC User Guide*.
- Verify that your security group has a rule that allows RDP access on port 3389.
- If you copied the password but get the error `Your credentials did not work`, try typing them manually when prompted. It's possible that you missed a character or got an extra white space character when you copied the password.
- Verify that the instance has passed status checks. For more information, see Status checks for Amazon EC2 instances and the section called “Linux instance failed status checks”.
- Verify that the route table for the subnet has a route that sends all traffic destined outside the VPC to the internet gateway for the VPC. For more information, see Creating a custom route table (Internet Gateways) in the *Amazon VPC User Guide*.
- Verify that Windows Firewall, or other firewall software, is not blocking RDP traffic to the instance. We recommend that you disable Windows Firewall and control access to your instance using security group rules. Try one of the following options:
  - Use AWSSupport-TroubleshootRDP to disable the Windows Firewall profiles using SSM Agent. This option requires that the instance is configured for AWS Systems Manager.
  - Use AWSSupport-ExecuteEC2Rescue.
  - Stop the instance, detach the root volume, and attach the volume to a temporary instance as a data volume. Connect to the temporary instance and bring the volume online. Load the registry hive from the data volume. Under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicyStandardProfile`, set `EnableFirewall` to 0. Unload the registry hive and then bring the volume offline. Detach the volume from the temporary instance and attach it to the original instance as the root volume. Start the original instance.
- Verify that Network Level Authentication is disabled on instances that are not part of an Active Directory domain (use AWSSupport-TroubleshootRDP to disable NLA).

- Verify that the Remote Desktop Service (TermService) Startup Type is Automatic and the service is started (use AWSSupport-TroubleshootRDP to enable and start the RDP service).
- Verify that you are connecting to the correct Remote Desktop Protocol port, which by default is 3389 (use AWSSupport-TroubleshootRDP to read the current RDP port and change it back to 3389).
- Verify that Remote Desktop connections are allowed on your instance (use AWSSupport-TroubleshootRDP to enable Remote Desktop connections).
- Verify that the password has not expired. If the password has expired, you can reset it. For more information, see Reset the Windows administrator password for an Amazon EC2 Windows instance.
- If you attempt to connect using a user that you created on the instance and receive the error The `user cannot connect to the server due to insufficient access privileges`, verify that you granted the user the right to log on locally. For more information, see Grant a Member the Right to Logon Locally.
- If you attempt more than the maximum allowed concurrent RDP sessions, your session is terminated with the message `Your Remote Desktop Services session has ended. Another user connected to the remote computer, so your connection was lost.` By default, you are allowed two concurrent RDP sessions to your instance.

Error using the macOS RDP client

If you are connecting to a Windows Server instance using the Remote Desktop Connection client from the Microsoft website, you may get the following error:

```
Remote Desktop Connection cannot verify the identity of the computer that you want to
 connect to.
```

Download the Microsoft Remote Desktop app from the Mac App Store and use the app to connect to your instance.

RDP displays a black screen instead of the desktop

Try the following to resolve this issue:

- Check the console output for additional information. To get the console output for your instance using the Amazon EC2 console, select the instance, and then choose **Actions**, **Monitor and troubleshoot**, **Get system log**.

- Verify that you are running the latest version of your RDP client.
- Try the default settings for the RDP client.
- If you are using Remote Desktop Connection, try starting it with the `/admin` option as follows.

```
mstsc /v:instance /admin
```

- If the server is running a full-screen application, it might have stopped responding. Use Ctrl +Shift+Esc to start Windows Task Manager, and then close the application.
- If the server is over-utilized, it might have stopped responding. To monitor the instance using the Amazon EC2 console, select the instance and then select the **Monitoring** tab. If you need to change the instance type to a larger size, see Amazon EC2 instance type changes.

Unable to remotely log on to an instance with a user that is not an administrator

If you are not able to remotely log on to a Windows instance with a user that is not an administrator account, ensure that you have granted the user the right to log on locally. See Grant a user or group the right to log on locally to the domain controllers in the domain.

Troubleshooting Remote Desktop issues using AWS Systems Manager

You can use AWS Systems Manager to troubleshoot issues connecting to your Windows instance using RDP.

AWSSupport-TroubleshootRDP

The AWSSupport-TroubleshootRDP automation document allows the user to check or modify common settings on the target instance that can impact Remote Desktop Protocol (RDP) connections, such as the **RDP Port**, **Network Layer Authentication (NLA)**, and **Windows Firewall** profiles. By default, the document reads and outputs the values of these settings.

The AWSSupport-TroubleshootRDP automation document can be used with EC2 instances, on-premises instances, and virtual machines (VMs) that are enabled for use with AWS Systems Manager (managed instances). In addition, it can also be used with EC2 instances for Windows Server that are *not* enabled for use with Systems Manager. For information about enabling instances for use with AWS Systems Manager, see Managed nodes in the*AWS Systems Manager User Guide*.

To troubleshoot using the AWSSupport-TroubleshootRDP document

1. Log in to the Systems Manager Console.
2. Verify that you are in the same Region as the impaired instance.
3. Choose **Documents** from the left navigation pane.
4. On the **Owned by Amazon** tab, enter `AWSSupport-TroubleshootRDP` in the search field. When the `AWSSupport-TroubleshootRDP` document appears, select it.
5. Choose **Execute automation**.
6. For **Execution Mode**, choose **Simple execution**.
7. For **Input parameters**, **InstanceId**, enable **Show interactive instance picker**.
8. Choose your Amazon EC2 instance.
9. Review the examples, then choose **Execute**.
10. To monitor the execution progress, for **Execution status**, wait for the status to change from **Pending** to **Success**. Expand **Outputs** to view the results. To view the output of individual steps, in **Executed Steps**, choose an item from **Step ID.**

AWSSupport-TroubleshootRDP examples

The following examples show you how to accomplish common troubleshooting tasks using AWSSupport-TroubleshootRDP. You can use either the example AWS CLI start-automation-execution command or the provided link to the AWS Management Console.

Example Example: Check the current RDP status

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom" --region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region#documentVersion=$LATEST
```

Example Example: Disable the Windows Firewall

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP"
 --parameters "InstanceId=i-1234567890abcdef0, Action=Custom, Firewall=Disable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&Firewall=Disable
```

Example Example: Disable Network Level Authentication

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, NLASettingAction=Disable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion
```

Example Example: Set RDP Service Startup Type to Automatic and start the RDP service

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPServiceStartupType=Auto,
 RDPServiceAction=Start" --region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/
AWSSupport-TroubleshootRDP?region=region_code#documentVersion=
$LATEST&RDPServiceStartupType=Auto&RDPServiceAction=Start
```

Example Example: Restore the default RDP Port (3389)

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RDPPortAction=Modify" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RDPPortAction=Modify
```

Example Example: Allow remote connections

AWS CLI:

```
aws ssm start-automation-execution --document-name "AWSSupport-TroubleshootRDP" --
parameters "InstanceId=i-1234567890abcdef0, Action=Custom, RemoteConnections=Enable" --
region region_code
```

AWS Systems Manager console:

```
https://console.aws.amazon.com/systems-manager/automation/execute/AWSSupport-
TroubleshootRDP?region=region_code#documentVersion=$LATEST&RemoteConnections=Enable
```

AWSSupport-ExecuteEC2Rescue

The AWSSupport-ExecuteEC2Rescue automation document uses EC2Rescue for Windows Server to automatically troubleshoot and restore EC2 instance connectivity and RDP issues. For more information, see Run the EC2Rescue tool on unreachable instances.

The AWSSupport-ExecuteEC2Rescue automation document requires a stop and restart of the instance. Systems Manager Automation stops the instance and creates an Amazon Machine Image (AMI). Data stored in instance store volumes is lost. The public IP address changes if you are not using an Elastic IP address. For more information, see Run the EC2Rescue tool on unreachable instances in the *AWS Systems Manager User Guide*.

To troubleshoot using the AWSSupport-ExecuteEC2Rescue document

1. Open the Systems Manager console.
2. Verify that you are in the same Region as the impaired Amazon EC2 instance.
3. In the navigation panel, choose **Documents**.

4. Search for and select the `AWSSupport-ExecuteEC2Rescue` document, and then choose **Execute automation**.
5. In **Execution Mode**, choose **Simple execution**.
6. In the **Input parameters** section, for **UnreachableInstanceId**, enter the Amazon EC2 instance ID of the unreachable instance.
7. (Optional) For **LogDestination**, enter the Amazon Simple Storage Service (Amazon S3) bucket name if you want to collect operating system logs for troubleshooting your Amazon EC2 instance. Logs are automatically uploaded to the specified bucket.
8. Choose **Execute**.
9. To monitor the execution progress, in **Execution status**, wait for the status to change from **Pending** to **Success**. Expand **Outputs** to view the results. To view the output of individual steps, in **Executed Steps**, choose the **Step ID**.

Enable Remote Desktop on an EC2 instance with remote registry

If your unreachable instance is not managed by AWS Systems Manager Session Manager, then you can use remote registry to enable Remote Desktop.

1. From the EC2 console, stop the unreachable instance.
2. Detach the root volume of the unreachable instance and attach it to a reachable instance in the same Availability Zone as a storage volume. If you don't have a reachable instance in the same Availability Zone, launch one. Note the device name of the root volume on the unreachable instance.
3. On the reachable instance, open Disk Management. You can do so by running the following command in the Command Prompt window.

   ```
   diskmgmt.msc
   ```
4. Right click the newly attached volume that came from the unreachable instance, and then choose **Online**.
5. Open the Windows Registry Editor. You can do so by running the following command in the Command Prompt window.

   ```
   regedit
   ```
6. In Registry Editor, choose **HKEY_LOCAL_MACHINE**, then select **File**, **Load Hive**.

7. Select the drive of the attached volume, navigate to `\Windows\System32\config\`, select SYSTEM, and then choose **Open**.
8. For **Key Name**, enter a unique name for the hive and choose **OK**.
9. Back up the registry hive before making any changes to the registry.
   a. In the Registry Editor console tree, select the hive that you loaded: **HKEY_LOCAL_MACHINE***your-key-name*.
   b. Choose **File**, **Export**.
   c. In the Export Registry File dialog box, choose the location to which you want to save the backup copy, and then type a name for the backup file in the **File name** field.
   d. Choose **Save**.
10. In Registry Editor, navigate to `HKEY_LOCAL_MACHINE\`*your key name*`\ControlSet001\Control\Terminal Server`, and then, in the details pane, double-click **fDenyTSConnections**.
11. In the **Edit DWORD** value box, enter `0` in the **Value data** field.
12. Choose **OK**.

> **Note**
>
> If the value in the **Value data** field is `1`, then the instance will deny remote desktop connections. A value of `0` allows remote desktop connections.

13. In Registry Editor, choose **HKEY_LOCAL_MACHINE***your-key-name*, then select **File**, **Unload Hive**.
14. Close Registry Editor and Disk Management.
15. From the EC2 console, detach the volume from the reachable instance and then reattach it to the unreachable instance. When attaching the volume to the unreachable instance, enter the device name that you saved earlier in the **device** field.
16. Restart the unreachable instance.

I've lost my private key. How can I connect to my Windows instance?

When you connect to a newly-launched Windows instance, you decrypt the password for the Administrator account using the private key for the key pair that you specified when you launched the instance.

If you lose the Administrator password and you no longer have the private key, you must reset the password or create a new instance. For more information, see Reset the Windows administrator password for an Amazon EC2 Windows instance. For steps to reset the password using an Systems Manager document, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

Troubleshoot Amazon EC2 Windows instance start issues

The following are troubleshooting tips to help you solve password and activation issues with Amazon EC2 Windows instances.

Issues

- "Password is not available"
- "Password not available yet"
- "Cannot retrieve Windows password"
- "Waiting for the metadata service"
- "Unable to activate Windows"
- "Windows is not genuine (0x80070005)"
- "No Terminal Server License Servers available to provide a license"
- "Some settings are managed by your organization"

"Password is not available"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

You can generate a password for the Administrator account for instances launched using a custom Windows AMI. To generate the password, you will need to configure some settings in the operating system before the AMI is created. For more information, see Create an Amazon EBS-backed AMI.

If your Windows instance isn't configured to generate a random password, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password is not available.
The instance was launched from a custom AMI, or the default password has changed. A
password cannot be retrieved for this instance. If you have forgotten your password,
 you can
reset it using the Amazon EC2 configuration service. For more information, see
 Passwords for a
Windows Server instance.
```

Check the console output for the instance to see whether the AMI that you used to launch it was created with password generation disabled. If password generation is disabled, the console output contains the following:

```
Ec2SetPassword: Disabled
```

If password generation is disabled and you don't remember the password for the original instance, you can reset the password for this instance. For more information, see Reset the Windows administrator password for an Amazon EC2 Windows instance.

"Password not available yet"

To connect to a Windows instance using Remote Desktop, you must specify an account and password. The accounts and passwords provided are based on the AMI that you used to launch the instance. You can either retrieve the auto-generated password for the Administrator account, or use the account and password that were in use in the original instance from which the AMI was created.

Your password should be available within a few minutes. If the password isn't available, you'll receive the following message when you retrieve the auto-generated password using the console:

```
Password not available yet.
Please wait at least 4 minutes after launching an instance before trying to retrieve
 the
auto-generated password.
```

If it's been longer than four minutes and you still can't get the password, it's possible that the launch agent for your instance is not configured to generate a password. Verify by checking whether the console output is empty. For more information, see Unable to get console output.

Also verify that the AWS Identity and Access Management (IAM) account being used to access the Management Portal has the `ec2:GetPasswordData` action allowed. For more information about IAM permissions, see What is IAM?.

"Cannot retrieve Windows password"

To retrieve the auto-generated password for the Administrator account, you must use the private key for the key pair that you specified when you launched the instance. If you didn't specify a key pair when you launched the instance, you'll receive the following message.

```
Cannot retrieve Windows password
```

You can terminate this instance and launch a new instance using the same AMI, making sure to specify a key pair.

"Waiting for the metadata service"

A Windows instance must obtain information from its instance metadata before it can activate itself. By default, the `WaitForMetaDataAvailable` setting ensures that the EC2Config service waits for the instance metadata to be accessible before continuing with the boot process. For more information, see Use instance metadata to manage your EC2 instance.

If the instance is failing the instance reachability test, try the following to resolve this issue.

- Check the CIDR block for your VPC. A Windows instance cannot boot correctly if it's launched into a VPC that has an IP address range from `224.0.0.0` to `255.255.255.255` (Class D and Class E IP address ranges). These IP address ranges are reserved, and should not be assigned to host devices. We recommend that you create a VPC with a CIDR block from the private (non-publicly routable) IP address ranges as specified in RFC 1918.
- It's possible that the system has been configured with a static IP address. Try creating a network interface and attaching it to the instance.
- **To enable DHCP on a Windows instance that you can't connect to**

  1. Stop the affected instance and detach its root volume.
  2. Launch a temporary instance in the same Availability Zone as the affected instance.

> **Warning**
>
> If your temporary instance is based on the same AMI that the original instance is based on, you must complete additional steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision. Alternatively, select a different AMI for the temporary instance. For example, if the original instance uses the AWS Windows AMI for Windows Server 2016, launch the temporary instance using the AWS Windows AMI for Windows Server 2019.

3. Attach the root volume from the affected instance to this temporary instance. Connect to the temporary instance, open the **Disk Management** utility, and bring the drive online.
4. From the temporary instance, open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SYSTEM`, and specify a key name when prompted (you can use any name).
5. Select the key that you just loaded and navigate to `ControlSet001\Services\Tcpip\Parameters\Interfaces`. Each network interface is listed by a GUID. Select the correct network interface. If DHCP is disabled and a static IP address assigned, `EnableDHCP` is set to 0. To enable DHCP, set `EnableDHCP` to 1, and delete the following keys if they exist: `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. Select the key again, and from the **File** menu, choose **Unload Hive**.

   > **Note**
   >
   > If you have multiple network interfaces, you'll need to identify the correct interface to enable DHCP. To identify the correct network interface, review the following key values `NameServer`, `SubnetMask`, `IPAddress`, and `DefaultGateway`. These values display the static configuration of the previous instance.

6. (Optional) If DHCP is already enabled, it's possible that you don't have a route to the metadata service. Updating EC2Config can resolve this issue.

   a. Download and install the latest version of the EC2Config service. For more information about installing this service, see the section called “Install EC2Config”.

   b. Extract the files from the `.zip` file to the `Temp` directory on the drive you attached.

c. Open **Regedit** and select **HKEY_LOCAL_MACHINE**. From the **File** menu, choose **Load Hive**. Select the drive, open the file `Windows\System32\config\SOFTWARE`, and specify a key name when prompted (you can use any name).

d. Select the key that you just loaded and navigate to `Microsoft\Windows\CurrentVersion`. Select the `RunOnce` key. (If this key doesn't exist, right-click `CurrentVersion`, point to **New**, select **Key**, and name the key `RunOnce`.) Right-click, point to **New**, and select **String Value**. Enter `Ec2Install` as the name and `C:\Temp\Ec2Install.exe -q` as the data.

e. Select the key again, and from the **File** menu, choose **Unload Hive**.

7. (Optional) If your temporary instance is based on the same AMI that the original instance is based on, you must complete the following steps or you won't be able to boot the original instance after you restore its root volume because of a disk signature collision.

> **⚠ Warning**
>
> The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see Configure the Registry.

a. Open a command prompt, type **regedit.exe**, and press Enter.

b. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.

c. Type **Windows Boot Manager** and then choose **Find Next**.

d. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.

e. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).

f. Locate the four-byte disk signature at offset 0x38 in the data. Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
```

```
...
```

g. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

h. Run the following DiskPart command to select the volume. (You can verify that the disk number is 1 using the **Disk Management** utility.)

```
DISKPART> select disk 1

Disk 1 is now the selected disk.
```

i. Run the following DiskPart command to get the disk signature.

```
DISKPART>  uniqueid disk

Disk ID: 0C764FA8
```

j. If the disk signature shown in the previous step doesn't match the disk signature from BCD that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

8. Using the **Disk Management** utility, bring the drive offline.

> **Note**
>
> The drive is automatically offline if the temporary instance is running the same operating system as the affected instance, so you won't need to bring it offline manually.

9. Detach the volume from the temporary instance. You can terminate the temporary instance if you have no further use for it.
10. Restore the root volume of the affected instance by attaching the volume as `/dev/sda1`.
11. Start the affected instance.

If you are connected to the instance, open an Internet browser from the instance and enter the following URL for the metadata server:

```
http://169.254.169.254/latest/meta-data/
```

If you can't contact the metadata server, try the following to resolve the issue:

- Download and install the latest version of the EC2Config service. For more information about installing this service, see the section called “Install EC2Config”.
- Check whether the Windows instance is running Red Hat PV drivers. If so, update to Citrix PV drivers. For more information, see the section called “Upgrade PV drivers”.
- Verify that the firewall, IPSec, and proxy settings do not block outgoing traffic to the metadata service (`169.254.169.254`) or the AWS KMS servers (the addresses are specified in `TargetKMSServer` elements in `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml`).
- Verify that you have a route to the metadata service (`169.254.169.254`) using the following command.

  ```
  route print
  ```

- Check for network issues that might affect the Availability Zone for your instance. Go to http://status.aws.amazon.com/.

"Unable to activate Windows"

Windows instances use Windows AWS KMS activation. You can receive this message: `A problem occurred when Windows tried to activate. Error Code 0xC004F074`, if your instance can't reach the AWS KMS server. Windows must be activated every 180 days. EC2Config attempts to contact the AWS KMS server before the activation period expires to ensure that Windows remains activated.

If you encounter a Windows activation issue, use the following procedure to resolve the issue.

For EC2Config (Windows Server 2012 R2 AMIs and earlier)

1. Download and install the latest version of the EC2Config service. For more information about installing this service, see the section called “Install EC2Config”.

2. Log onto the instance and open the following file: `C:\Program Files\Amazon\Ec2ConfigService\Settings\config.xml`.
3. Locate the **Ec2WindowsActivate** plugin in the `config.xml` file. Change the state to **Enabled** and save your changes.
4. In the Windows Services snap-in, restart the EC2Config service or reboot the instance.

If this does not resolve the activation issue, follow these additional steps.

1. Set the AWS KMS target: **C:\> slmgr.vbs /skms 169.254.169.250:1688**
2. Activate Windows: **C:\> slmgr.vbs /ato**

For EC2Launch (Windows Server 2016 AMIs and later)

1. From a PowerShell prompt with administrative rights, import the EC2Launch module:

```
PS C:\> Import-Module "C:\ProgramData\Amazon\EC2-Windows\Launch\Module
\Ec2Launch.psd1"
```

2. Call the Add-Routes function to see the list of new routes:

```
PS C:\> Add-Routes
```

3. Call the Set-ActivationSettings function:

```
PS C:\> Set-Activationsettings
```

4. Then, run the following script to activate Windows:

```
PS C:\> cscript "${env:SYSTEMROOT}\system32\slmgr.vbs" /ato
```

For both EC2Config and EC2Launch, if you are still receiving an activation error, verify the following information.

- Verify that you have routes to the AWS KMS servers. Open `C:\Program Files\Amazon\Ec2ConfigService\Settings\ActivationSettings.xml` and locate the `TargetKMSServer` elements. Run the following command and check whether the addresses for these AWS KMS servers are listed.

```
route print
```

- Verify that the AWS KMS client key is set. Run the following command and check the output.

  ```
  C:\Windows\System32\slmgr.vbs /dlv
  ```

  If the output contains Error: product key not found, the AWS KMS client key isn't set. If the AWS KMS client key isn't set, look up the client key as described in this Microsoft article: AWS KMS client activation and product keys, and then run the following command to set the AWS KMS client key.

  ```
  C:\Windows\System32\slmgr.vbs /ipk client_key
  ```

- Verify that the system has the correct time and time zone. If you are using a time zone other than UTC, add the following registry key and set it to 1 to ensure that the time is correct: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal`.
- If Windows Firewall is enabled, temporarily disable it using the following command.

  ```
  netsh advfirewall set allprofiles state off
  ```

"Windows is not genuine (0x80070005)"

Windows instances use Windows AWS KMS activation. If an instance is unable to complete the activation process, it reports that the copy of Windows is not genuine.

Try the suggestions for "Unable to activate Windows".

"No Terminal Server License Servers available to provide a license"

By default, Windows Server is licensed for two simultaneous users through Remote Desktop. If you need to provide more than two users with simultaneous access to your Windows instance through Remote Desktop, you can purchase a Remote Desktop Services client access license (CAL) and install the Remote Desktop Session Host and Remote Desktop Licensing Server roles.

Check for the following issues:

- You've exceeded the maximum number of concurrent RDP sessions.

- You've installed the Windows Remote Desktop Services role.
- Licensing has expired. If the licensing has expired, you can't connect to your Windows instance as a user. You can try the following:
  - Connect to the instance from the command line using an `/admin` parameter, for example:

    ```
    mstsc /v:instance /admin
    ```

    For more information, see the following Microsoft article: Access Remote Desktop Via Command Line.
  - Stop the instance, detach its Amazon EBS volumes, and attach them to another instance in the same Availability Zone to recover your data.

"Some settings are managed by your organization"

Instances launched from the latest Windows Server AMIs might show a Windows Update dialog message stating "Some settings are managed by your organization." This message appears as a result of changes in Windows Server and does not impact the behavior of Windows Update or your ability to manage update settings.

To remove the warning

1. Open `gpedit.msc` and navigate to **Computer Configuration**, **Administrative Templates**, **Windows Components**, **Windows updates**. Edit **Configure Automatic Update**, and set it to **enabled**.
2. In a command prompt, update group policy using **gpupdate /force**.
3. Close and reopen the Windows Update Settings. You will see the above message about your settings being managed by your organization, followed by "We'll automatically download updates, except on metered connections (where charges may apply). In that case, we'll automatically download those updates required to keep Windows running smoothly."
4. Return to `gpedit.msc` and set the group policy back to **not configured**. Run **gpupdate /force** again.
5. Close the command prompt and wait a few minutes.
6. Reopen the Windows Update Settings. You should not see the message "Some settings are managed by your organization."

Troubleshoot issues with Amazon EC2 Windows instances

The following are troubleshooting tips to help you solve the issues with Amazon EC2 Windows instances.

Issues

Unable to connect AWS Systems Manager Sessions Manager to a Windows Server 2025 instance

You may encounter an issue connecting AWS Systems Manager Sessions Manager to a Windows Server 2025 instance. To address this issue, log onto the instance, then navigate to `Settings > Apps > Optional Features`, and add `WMIC`. Restart the SSM Agent service or reboot the instance, and Sessions Manager should connect.

You can also use the following PowerShell command to perform the same action:

```
Start-Process -FilePath "$env:SystemRoot\system32\Dism.exe" -ArgumentList @('/
Online', '/Add-Capability', '/CapabilityName:WMIC~~~~') -Wait; Restart-Service -Name
 AmazonSSMAgent
```

EBS volumes don't initialize on Windows Server 2016 and 2019

Instances created from Amazon Machine Images (AMIs) for Windows Server 2016 and 2019 use the EC2Launch v1 agent for a variety of startup tasks, including initializing EBS volumes. By default, EC2Launch v1 doesn't initialize secondary volumes. However, you can configure EC2Launch v1 to initialize these disks automatically, as follows.

Map drive letters to volumes

1. Connect to the instance to configure and open the `C:\ProgramData\Amazon\EC2-Windows\Launch\Config\DriveLetterMappingConfig.json` file in a text editor.
2. Specify the volume settings as follows:

```
{
"driveLetterMapping": [
 {
   "volumeName": "sample volume",
   "driveLetter": "H"
 }]
}
```

3. Save your changes and close the file.
4. Open Windows PowerShell and use the following command to run the EC2Launch v1 script that initializes the disks:

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1
```

To initialize the disks each time the instance boots, add the `-Schedule` flag as follows:

```
PS C:\>  C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeDisks.ps1 -
Schedule
```

The EC2Launch v1 agent can run instance initialization scripts such as `initializeDisks.ps1` in parallel with the `InitializeInstance.ps1` script. If the `InitializeInstance.ps1` script reboots the instance, it might interrupt other scheduled tasks that run at instance startup. To avoid any potential conflicts, we recommend that you add logic to your `initializeDisks.ps1` script to ensure that instance initialization has finished first.

> **Note**
>
> If the EC2Launch script does not initialize the volumes, ensure that the volumes are online. If the volumes are offline, run the following command to bring all disks online.

```
PS C:\> Get-Disk | Where-Object IsOffline -Eq $True | Set-Disk -IsOffline
 $False
```

Boot an EC2 Windows instance into Directory Services Restore Mode (DSRM)

If an instance running Microsoft Active Directory experiences a system failure or other critical issues you can troubleshoot the instance by booting into a special version of Safe Mode called *Directory Services Restore Mode* (DSRM). In DSRM you can repair or recover Active Directory.

Driver support for DSRM

How you enable DSRM and boot into the instance depends on the drivers the instance is running. In the EC2 console you can view driver version details for an instance from the System Log. The following table shows which drivers are supported for DSRM.

Driver Versions	DSRM Supported?	Next Steps
Citrix PV 5.9	No	Restore the instance from a backup. You cannot enable DSRM.
AWS PV 7.2.0	No	Though DSRM is not supported for this driver, you can still detach the root volume from the instance, take a snapshot of the volume or create an AMI from it, and attach it to another instance in the same Availability Zone as a secondary volume. You can then enable DSRM (as described in this section).
AWS PV 7.2.2 and later	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).
Enhanced Networking	Yes	Detach the root volume, attach it to another instance, and enable DSRM (as described in this section).

For information about how to enable enhanced networking, see the section called "Elastic Network Adapter (ENA)". For information about upgrading AWS PV drivers, see Upgrade PV drivers on Windows instances.

Configure an instance to boot into DSRM

EC2 Windows instances do not have network connectivity before the operating system is running. For this reason, you cannot press the F8 button on your keyboard to select a boot option. You must use one of the following procedures to boot an EC2 Windows Server instance into DSRM.

If you suspect that Active Directory has been corrupted and the instance is still running, you can configure the instance to boot into DSRM using either the System Configuration dialog box or the command prompt.

To boot an online instance into DSRM using the System Configuration dialog box

1. In the **Run** dialog box, type `msconfig` and press Enter.
2. Choose the **Boot** tab.
3. Under **Boot options** choose **Safe boot**.
4. Choose **Active Directory repair** and then choose **OK**. The system prompts you to reboot the server.

To boot an online instance into DSRM using the command line

From a Command Prompt window, run the following command:

```
bcdedit /set safeboot dsrepair
```

If an instance is offline and unreachable, you must detach the root volume and attach it to another instance to enable DSRM mode.

To boot an offline instance into DSRM

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Locate and select the affected instance. Choose **Instance state**, **Stop instance**.
4. Choose **Launch instances** and create a temporary instance in the same Availability Zone as the affected instance. Choose an instance type that uses a different version of Windows.

For example, if your instance is Windows Server 2016, then choose a Windows Server 2019 instance.

> **Important**
>
> If you do not create the instance in the same Availability Zone as the affected instance you will not be able to attach the root volume of the affected instance to the new instance.

5. In the navigation pane, choose **Volumes**.
6. Locate the root volume of the affected instance. Detach the volume and attach it to the temporary instance you created earlier. Attach it with the default device name (xvdf).
7. Use Remote Desktop to connect to the temporary instance, and then use the Disk Management utility to make the volume available for use.
8. Open a command prompt and run the following command. Replace *D* with the actual drive letter of the secondary volume you just attached:

```
bcdedit /store D:\Boot\BCD /set {default} safeboot dsrepair
```

9. In the Disk Management Utility, choose the drive you attached earlier, open the context (right-click) menu, and choose **Offline**.
10. In the EC2 console, detach the affected volume from the temporary instance and reattach it to your original instance with the device name `/dev/sda1`. You must specify this device name to designate the volume as a root volume.
11. Start the instance.
12. After the instance passes the health checks in the EC2 console, connect to the instance using Remote Desktop and verify that it boots into DSRM mode.
13. (Optional) Delete or stop the temporary instance you created in this procedure.

Instance loses network connectivity or scheduled tasks don't run when expected

If you restart your instance and it loses network connectivity, it's possible that the instance has the wrong time.

By default, Windows instances use Coordinated Universal Time (UTC). If you set the time for your instance to a different time zone and then restart it, the time becomes offset and the instance temporarily loses its IP address. The instance regains network connectivity eventually, but this can take several hours. The amount of time that it takes for the instance to regain network connectivity depends on the difference between UTC and the other time zone.

This same time issue can also result in scheduled tasks not running when you expect them to. In this case, the scheduled tasks do not run when expected because the instance has the incorrect time.

To use a time zone other than UTC persistently, you must set the **RealTimeIsUniversal** registry key. Without this key, an instance uses UTC after you restart it.

To resolve time issues that cause a loss of network connectivity

1. Ensure that you are running the recommended PV drivers. For more information, see the section called “Upgrade PV drivers”.
2. Verify that the following registry key exists and is set to 1: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal**

Unable to get console output

For Windows instances, the instance console displays the output from tasks performed during the Windows boot process. If Windows boots successfully, the last message logged is `Windows is Ready to use`. You can also display event log messages in the console, but this feature might not be enabled by default depending on your version of Windows. For more information, see the section called “Windows launch agents”.

To get the console output for your instance using the Amazon EC2 console, select the instance, and then choose **Actions**, **Monitor and troubleshoot**, **Get system log**. To get the console output using the command line, use one of the following commands: get-console-output (AWS CLI) or Get-EC2ConsoleOutput (AWS Tools for Windows PowerShell).

For instances running Windows Server 2012 R2 and earlier, if the console output is empty, it could indicate an issue with the EC2Config service, such as a misconfigured configuration file, or that Windows failed to boot properly. To fix the issue, download and install the latest version of EC2Config. For more information, see the section called “Install EC2Config”.

Windows Server 2012 R2 not available on the network

For information about troubleshooting a Windows Server 2012 R2 instance that is not available on the network, see Windows Server 2012 R2 loses network and storage connectivity after an instance reboot.

Disk signature collision

You can check for and resolve disk signature collisions using EC2Rescue for Windows Server. Or, you can manually resolve disk signature issues by performing the following steps.

> **Warning**
>
> The following procedure describes how to edit the Windows Registry using Registry Editor. If you are not familiar with the Windows Registry or how to safely make changes using Registry Editor, see Configure the Registry.

1. Open a command prompt, type **regedit.exe**, and press Enter.
2. In the **Registry Editor**, choose **HKEY_LOCAL_MACHINE** from the context menu (right-click), and then choose **Find**.
3. Type **Windows Boot Manager** and then choose **Find Next**.
4. Choose the key named `11000001`. This key is a sibling of the key you found in the previous step.
5. In the right pane, choose `Element` and then choose **Modify** from the context menu (right-click).
6. Locate the four-byte disk signature at offset 0x38 in the data. This is the Boot Configuration Database signature (BCD). Reverse the bytes to create the disk signature, and write it down. For example, the disk signature represented by the following data is `E9EB3AA5`:

```
...
0030  00 00 00 00 01 00 00 00
0038  A5 3A EB E9 00 00 00 00
0040  00 00 00 00 00 00 00 00
...
```

7. In a Command Prompt window, run the following command to start Microsoft DiskPart.

```
diskpart
```

8. Run the `select disk` DiskPart command and specify the disk number for the volume with the disk signature collision.

   Tip

   To check the disk number for the volume with the disk signature collision, use the **Disk Management** utility. Open a command prompt, type `compmgmt.msc` and press **Enter**. In the left-hand navigation panel, double-click **Disk Management**. In the **Disk Management** utility, check the disk number for the offline volume with the disk signature collision.

```
DISKPART> select disk 1
Disk 1 is now the selected disk.
```

9. Run the following DiskPart command to get the disk signature.

```
DISKPART>  uniqueid disk
Disk ID: 0C764FA8
```

10. If the disk signature shown in the previous step doesn't match the disk signature that you wrote down earlier, use the following DiskPart command to change the disk signature so that it matches:

```
DISKPART> uniqueid disk id=E9EB3AA5
```

Reset the Windows administrator password for an Amazon EC2 Windows instance

If you are no longer able to connect to your Amazon EC2 Windows instance because the Windows administrator password is lost or expired, you can reset the password.

> **Note**
>
> There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

The manual methods to reset the administrator password use EC2Launch v2, EC2Config, or EC2Launch.

- For all supported Windows AMIs that include the EC2Launch v2 agent, use EC2Launch v2.
- For Windows AMIs before Windows Server 2016, use the EC2Config service.
- For Windows Server 2016 and later AMIs, use the EC2Launch service.

These procedures also describe how to connect to an instance if you lost the key pair that was used to create the instance. Amazon EC2 uses a public key to encrypt a piece of data, such as a password, and a private key to decrypt the data. The public and private keys are known as a *key pair*. With Windows instances, you use a key pair to obtain the administrator password and then log in using RDP.

> **Note**
>
> If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Use EC2Rescue for Windows Server with Systems Manager Run Command.

Contents

Reset Windows admin password for EC2 instance using EC2Launch v2

If you have lost your Windows administrator password and are using a supported Windows AMI that includes the EC2Launch v2 agent, you can use EC2Launch v2 to generate a new password.

If you are using a Windows Server 2016 or later AMI that does not include the EC2Launch v2 agent, see Reset Windows admin password for EC2 instance using EC2Launch.

If you are using a Windows Server AMI earlier than Windows Server 2016 that does not include the EC2Launch v2 agent, see Reset Windows admin password for EC2 instance using EC2Config.

Note

If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Use EC2Rescue for Windows Server with Systems Manager Run Command.

Note

There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Launch v2, you need to do the following:

- Step 1: Verify that the EC2Launch v2 agent is running
- Step 2: Detach the root volume from the instance
- Step 3: Attach the volume to a temporary instance
- Step 4: Delete the .run-once file
- Step 5: Restart the original instance

Step 1: Verify that the EC2Launch v2 agent is running

Before you attempt to reset the administrator password, verify that the EC2Launch v2 agent is installed and running. You use the EC2Launch v2 agent to reset the administrator password later in this section.

To verify that the EC2Launch v2 agent is running

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then select the instance that requires a password reset. This instance is referred to as the *original* instance in this procedure.
3. Choose **Actions**, **Monitor and troubleshoot**, **Get system log**.
4. Locate the EC2 Launch entry, for example, **Launch: EC2Launch v2 service v2.0.124**. If you see this entry, the EC2Launch v2 service is running.

   If the system log output is empty, or if the EC2Launch v2 agent is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see Capture a screenshot of an unreachable instance.

Step 2: Detach the root volume from the instance

You can't use EC2Launch v2 to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance that requires a password reset and choose **Instance state**, **Stop instance**. After the status of the instance changes to **Stopped**, continue with the next step.
4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.

   a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.

b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.

c. With the instance selected, choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**.

d. In the navigation pane, choose **AMIs**. Wait for the image status to change to **available**. Then, select the image and choose **Launch instance from AMI**.

e. Complete the fields to launch an instance, making sure to select the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch instance**.

f. When prompted, choose the key pair that you created for the new instance, and then choose **Launch instance**.

g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.

5. Detach the root volume from the original instance as follows:

a. Select the original instance and choose the **Storage** tab. Note the name of the root device under **Root device name**. Find the volume with this device name under **Block devices**, and note the volume ID.

b. In the navigation pane, choose **Volumes**.

c. In the list of volumes, select the volume that you noted as the root device, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

Step 3: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to modify the configuration file.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:

a. In the navigation pane, choose **Instances**, choose **Launch instances**, and then select an AMI.

> **Important**
>
> To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2019, launch the temporary instance using the base AMI for Windows Server 2016.

b. Leave the default instance type and choose **Next: Configure Instance Details**.

c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

> **Important**
>
> The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.

d. On the **Review Instance Launch** page, choose **Launch**.

e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

2. Attach the volume to the temporary instance as a secondary volume as follows:

a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions**, **Attach Volume**.

b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.

c. For **Device**, type **`xvdf`** (if it isn't already there), and choose **Attach**.

Step 4: Delete the .run-once file

You must now delete the `.run-once` file from the offline volume attached to the instance. This directs EC2Launch v2 to run all tasks with a frequency of `once`, which includes setting the administrator password. The file path in the secondary volume that you attached will be similar to `D:\ProgramData\Amazon\EC2Launch\state\.run-once`.

To delete the .run-once file

1. Open the **Disk Management** utility, and bring the drive online using these instructions: Make an Amazon EBS volume available for use.
2. Locate the `.run-once` file in the disk you brought online.
3. Delete the .run-once file.

> **Important**
>
> Any scripts set to run once will be triggered by this action.

Step 5: Restart the original instance

After you have deleted the `.run-once` file, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

1. Reattach the volume to the original instance as follows:
   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions**, **Attach Volume**.
   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.
   c. For **Device**, type **`/dev/sda1`**.
   d. Choose **Attach**. After the volume status changes to `in-use`, continue to the next step.
2. In the navigation pane, choose **Instances**. Select the original instance and choose **Instance state**, **Start instance**. After the instance state changes to `Running`, continue to the next step.
3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see Connect to your Windows instance using RDP.

   > **Important**
   >
   > The instance gets a new public IP address after you stop and start it. Make sure to connect to the instance using its current public DNS name. For more information, see Amazon EC2 instance state changes.

4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State**, **Terminate instance**.

Reset Windows admin password for EC2 instance using EC2Launch

If you have lost your Windows administrator password and are using a Windows Server 2016 or later AMI, you can use the EC2Rescue tool, which uses the EC2Launch service to generate a new password.

If you are using a Windows Server 2016 or later AMI that does not include the EC2Launch v2 agent, you can use EC2Launch v2 to generate a new password.

If you are using a Windows Server AMI earlier than Windows Server 2016, see Reset Windows admin password for EC2 instance using EC2Config.

Warning

When you stop an instance, the data on any instance store volumes is erased. To keep data from instance store volumes, be sure to back it up to persistent storage.

Note

If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Use EC2Rescue for Windows Server with Systems Manager Run Command.

Note

There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Launch, you need to do the following:

Step 1: Detach the root volume from the instance

You can't use EC2Launch to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance that requires a password reset and choose **Instance state**, **Stop instance**. After the status of the instance changes to **Stopped**, continue with the next step.
4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.

    a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.

    b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.

    c. With the instance selected, choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**.

    d. In the navigation pane, choose **AMIs**. Wait for the image status to change to **available**. Then, select the image and choose **Launch instance from AMI**.

    e. Complete the fields to launch an instance, making sure to select the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch instance**.

    f. When prompted, choose the key pair that you created for the new instance, and then choose **Launch instance**.

g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.

5. Detach the root volume from the original instance as follows:

   a. Select the original instance and choose the **Storage** tab. Note the name of the root device under **Root device name**. Find the volume with this device name under **Block devices**, and note the volume ID.

   b. In the navigation pane, choose **Volumes**.

   c. In the list of volumes, select the volume that you noted as the root device, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

Step 2: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to run EC2Launch.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:

   a. In the navigation pane, choose **Instances**, choose **Launch instances**, and then select an AMI.

   > **Important**
   >
   > To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2019, launch the temporary instance using the base AMI for Windows Server 2016.

   b. Leave the default instance type and choose **Next: Configure Instance Details**.

   c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

> **Important**
>
> The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.

d. On the **Review Instance Launch** page, choose **Launch**.

e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

2. Attach the volume to the temporary instance as a secondary volume as follows:

a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions**, **Attach Volume**.

b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.

c. For **Device**, type **xvdf** (if it isn't already there), and choose **Attach**.

Step 3: Reset the administrator password

Next, connect to the temporary instance and use EC2Launch to reset the administrator password.

To reset the administrator password

1. Connect to the temporary instance and use the EC2Rescue for Windows Server tool on the instance to reset the administrator password as follows:

a. Download the EC2Rescue for Windows Server zip file, extract the contents, and run **EC2Rescue.exe**.

b. On the **License Agreement** screen, read the license agreement, and, if you accept the terms, choose **I Agree**.

c. On the **Welcome to EC2Rescue for Windows Server** screen, choose **Next**.

d. On the **Select mode** screen, choose **Offline instance**.

e. On the **Select a disk** screen, select the **xvdf** device and choose **Next**.

f. Confirm the disk selection and choose **Yes**.

g. After the volume has loaded, choose **OK**.

h. On the **Select Offline Instance Option** screen, choose **Diagnose and Rescue**.

i. On the **Summary** screen, review the information and choose **Next**.

j. On the **Detected possible issues** screen, select **Reset Administrator Password** and choose **Next**.

k. On the **Confirm** screen, choose **Rescue**, **OK**.

l. On the **Done** screen, choose **Finish**.

m. Close the EC2Rescue for Windows Server tool, disconnect from the temporary instance, and then return to the Amazon EC2 console.

2. Detach the secondary (`xvdf`) volume from the temporary instance as follows:

a. In the navigation pane, choose **Instances** and select the temporary instance.

b. On the **Storage** tab for the temporary instance, note the ID of the EBS volume listed as **xvdf**.

c. In the navigation pane, choose **Volumes**.

d. In the list of volumes, select the volume noted in the previous step, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

Step 4: Restart the original instance

After you have reset the administrator password using EC2Launch, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

To restart the original instance

1. Reattach the volume to the original instance as follows:

a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions**, **Attach Volume**.

b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.

c. For **Device**, type **`/dev/sda1`**.

d. Choose **Attach**. After the volume status changes to `in-use`, continue to the next step.

2. In the navigation pane, choose **Instances**. Select the original instance and choose **Instance state**, **Start instance**. After the instance state changes to `Running`, continue to the next step.
3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see Connect to your Windows instance using RDP.
4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State**, **Terminate instance**.

Reset Windows admin password for EC2 instance using EC2Config

If you have lost your Windows administrator password and are using a Windows AMI before Windows Server 2016, you can use the EC2Config agent to generate a new password.

If you are using a Windows Server 2016 or later AMI, see Reset Windows admin password for EC2 instance using EC2Launch or, you can use the EC2Rescue tool, which uses the EC2Launch service to generate a new password.

> **Note**
>
> If you have disabled the local administrator account on the instance and your instance is configured for Systems Manager, you can also re-enable and reset your local administrator password by using EC2Rescue and Run Command. For more information, see Use EC2Rescue for Windows Server with Systems Manager Run Command.

> **Note**
>
> There is an AWS Systems Manager Automation document that automatically applies the manual steps necessary to reset the local administrator password. For more information, see Reset passwords and SSH keys on EC2 instances in the *AWS Systems Manager User Guide*.

To reset your Windows administrator password using EC2Config, you need to do the following:

- Step 1: Verify that the EC2Config service is running
- Step 2: Detach the root volume from the instance

- Step 3: Attach the volume to a temporary instance
- Step 4: Modify the configuration file
- Step 5: Restart the original instance

Step 1: Verify that the EC2Config service is running

Before you attempt to reset the administrator password, verify that the EC2Config service is installed and running. You use the EC2Config service to reset the administrator password later in this section.

To verify that the EC2Config service is running

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances** and then select the instance that requires a password reset. This instance is referred to as the *original* instance in this procedure.
3. Choose **Actions**, **Monitor and troubleshoot**, **Get system log**.
4. Locate the EC2 Agent entry, for example, **EC2 Agent: Ec2Config service v3.18.1118**. If you see this entry, the EC2Config service is running.

   If the system log output is empty, or if the EC2Config service is not running, troubleshoot the instance using the Instance Console Screenshot service. For more information, see Capture a screenshot of an unreachable instance.

Step 2: Detach the root volume from the instance

You can't use EC2Config to reset an administrator password if the volume on which the password is stored is attached to an instance as the root volume. You must detach the volume from the original instance before you can attach it to a temporary instance as a secondary volume.

To detach the root volume from the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance that requires a password reset and choose **Instance state**, **Stop instance**. After the status of the instance changes to **Stopped**, continue with the next step.

4. (Optional) If you have the private key that you specified when you launched this instance, continue with the next step. Otherwise, use the following steps to replace the instance with a new instance that you launch with a new key pair.

   a. Create a new key pair using the Amazon EC2 console. To give your new key pair the same name as the one for which you lost the private key, you must first delete the existing key pair.

   b. Select the instance to replace. Note the instance type, VPC, subnet, security group, and IAM role of the instance.

   c. With the instance selected, choose **Actions**, **Image and templates**, **Create image**. Type a name and a description for the image and choose **Create image**.

   d. In the navigation pane, choose **AMIs**. Wait for the image status to change to **available**. Then, select the image and choose **Launch instance from AMI**.

   e. Complete the fields to launch an instance, making sure to select the same instance type, VPC, subnet, security group, and IAM role as the instance to replace, and then choose **Launch instance**.

   f. When prompted, choose the key pair that you created for the new instance, and then choose **Launch instance**.

   g. (Optional) If the original instance has an associated Elastic IP address, transfer it to the new instance. If the original instance has EBS volumes in addition to the root volume, transfer them to the new instance.

5. Detach the root volume from the original instance as follows:

   a. Select the original instance and choose the **Storage** tab. Note the name of the root device under **Root device name**. Find the volume with this device name under **Block devices**, and note the volume ID.

   b. In the navigation pane, choose **Volumes**.

   c. In the list of volumes, select the volume that you noted as the root device, and choose **Actions**, **Detach Volume**. After the volume status changes to **available**, continue with the next step.

6. If you created a new instance to replace your original instance, you can terminate the original instance now. It's no longer needed. For the remainder of this procedure, all references to the original instance apply to the new instance that you created.

Step 3: Attach the volume to a temporary instance

Next, launch a temporary instance and attach the volume to it as a secondary volume. This is the instance you use to modify the configuration file.

To launch a temporary instance and attach the volume

1. Launch the temporary instance as follows:

   a. In the navigation pane, choose **Instances**, choose **Launch instances**, and then select an AMI.

   > **Important**
   >
   > To avoid disk signature collisions, you must select an AMI for a different version of Windows. For example, if the original instance runs Windows Server 2019, launch the temporary instance using the base AMI for Windows Server 2016.

   b. Leave the default instance type and choose **Next: Configure Instance Details**.

   c. On the **Configure Instance Details** page, for **Subnet**, select the same Availability Zone as the original instance and choose **Review and Launch**.

   > **Important**
   >
   > The temporary instance must be in the same Availability Zone as the original instance. If your temporary instance is in a different Availability Zone, you can't attach the original instance's root volume to it.

   d. On the **Review Instance Launch** page, choose **Launch**.

   e. When prompted, create a new key pair, download it to a safe location on your computer, and then choose **Launch Instances**.

2. Attach the volume to the temporary instance as a secondary volume as follows:

   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the original instance, and then choose **Actions**, **Attach Volume**.

   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your temporary instance and select the instance from the list.

   c. For **Device**, type **xvdf** (if it isn't already there), and choose **Attach**.

Step 4: Modify the configuration file

After you have attached the volume to the temporary instance as a secondary volume, modify the `Ec2SetPassword` plugin in the configuration file.

To modify the configuration file

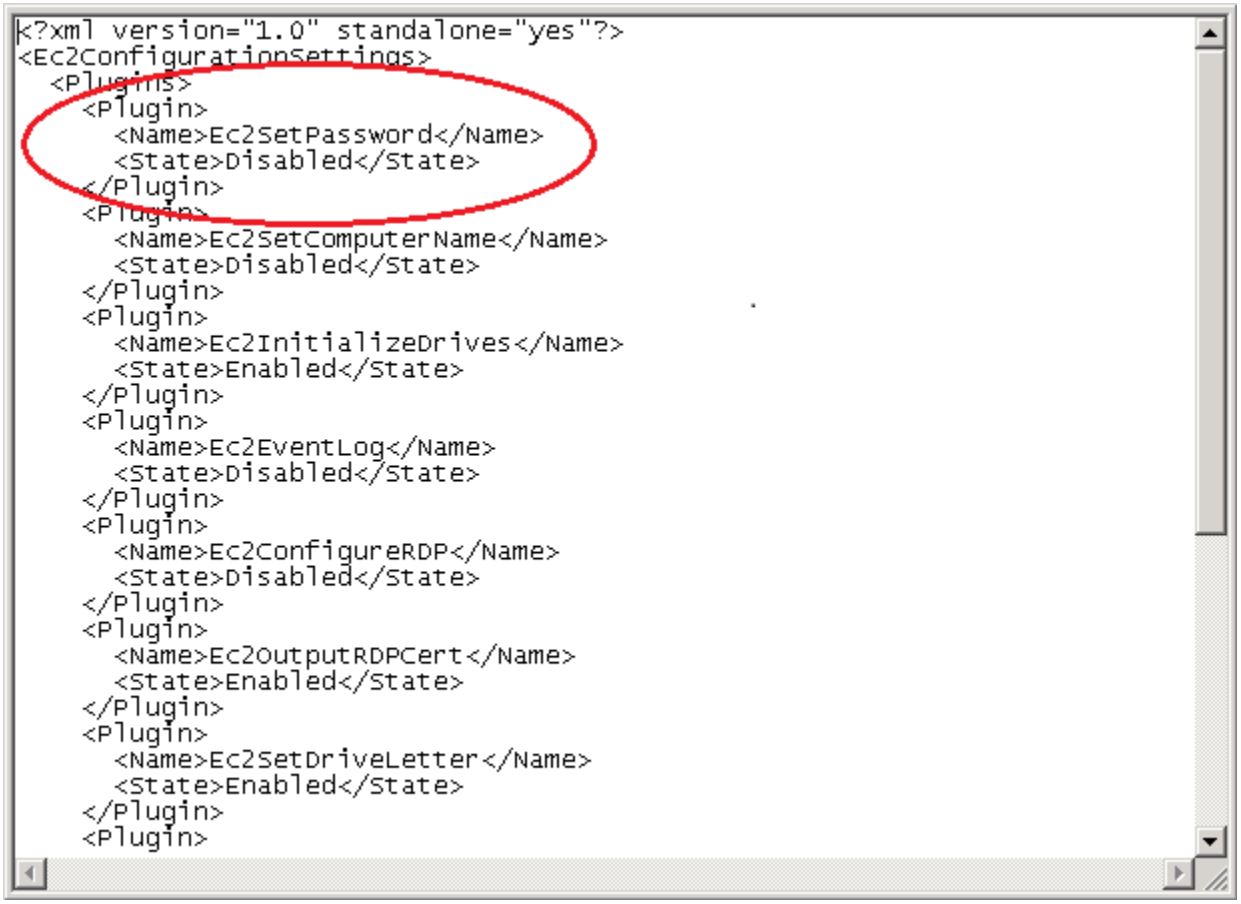
1. From the temporary instance, modify the configuration file on the secondary volume as follows:

   a. Launch and connect to the temporary instance.

   b. Use the following instructions to bring the drive online: Make an Amazon EBS volume available for use.

   c. Navigate to the secondary volume, and open `\Program Files\Amazon \Ec2ConfigService\Settings\config.xml` using a text editor, such as Notepad.

   d. At the top of the file, find the plugin with the name `Ec2SetPassword`, as shown in the screenshot. Change the state from `Disabled` to `Enabled` and save the file.



2. After you have modified the configuration file, detach the secondary volume from the temporary instance as follows:

   a. Using the **Disk Management** utility, bring the volume offline.

   b. Disconnect from the temporary instance and return to the Amazon EC2 console.

   c. In the navigation pane, choose **Volumes**, select the volume, and then choose **Actions**, **Detach Volume**. After the volume's status changes to **available**, continue with the next step.

Step 5: Restart the original instance

After you have modified the configuration file, reattach the volume to the original instance as the root volume and connect to the instance using its key pair to retrieve the administrator password.

1. Reattach the volume to the original instance as follows:

   a. In the navigation pane, choose **Volumes**, select the volume that you detached from the temporary instance, and then choose **Actions**, **Attach Volume**.

   b. In the **Attach Volume** dialog box, for **Instances**, start typing the name or ID of your original instance and then select the instance.

   c. For **Device**, type **`/dev/sda1`**.

   d. Choose **Attach**. After the volume status changes to `in-use`, continue to the next step.

2. In the navigation pane, choose **Instances**. Select the original instance and choose **Instance state**, **Start instance**. After the instance state changes to `Running`, continue to the next step.

3. Retrieve your new Windows administrator password using the private key for the new key pair and connect to the instance. For more information, see Connect to your Windows instance using RDP.

   > **Important**
   >
   > The instance gets a new public IP address after you stop and start it. Make sure to connect to the instance using its current public DNS name. For more information, see Amazon EC2 instance state changes.

4. (Optional) If you have no further use for the temporary instance, you can terminate it. Select the temporary instance, and choose **Instance State**, **Terminate instance**.

Troubleshoot Sysprep issues with Amazon EC2 Windows instances

If you experience problems or receive error messages during image preparations, review the following logs. Log location varies depending on whether you are running EC2Config, EC2Launch v1, or EC2Launch v2 with Sysprep.

- `%WINDIR%\Panther\Unattendgc` (EC2Config, EC2Launch v1, and EC2Launch v2)
- `%WINDIR%\System32\Sysprep\Panther` (EC2Config, EC2Launch v1, and EC2Launch v2)
- `C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt` (EC2Config only)
- `C:\ProgramData\Amazon\Ec2Config\Logs` (EC2Config only)
- `C:\ProgramData\Amazon\EC2-Windows\Launch\Log\EC2Launch.log` (EC2Launch v1 only)
- `%ProgramData%\Amazon\EC2Launch\log\agent.log` (EC2Launch v2 only)

If you receive an error message during image preparation with Sysprep, the OS might not be reachable. To review the log files, you must stop the instance, attach its root volume to another healthy instance as a secondary volume, and then review the logs mentioned earlier on the secondary volume. For more information about the purpose of the log files by name, see Windows Setup-Related Log Files in the Microsoft documentation.

If you locate errors in the Unattendgc log file, use the Microsoft Error Lookup Tool to get more details about the error. The following issue reported in the Unattendgc log file is typically the result of one or more corrupted user profiles on the instance:

```
Error [Shell Unattend] _FindLatestProfile failed (0x80070003) [gle=0x00000003]
Error [Shell Unattend] CopyProfile failed (0x80070003) [gle=0x00000003]
```

There are two options for resolving this issue:

Option 1

Use Regedit on the instance to search for the following key. Verify that there are no profile registry keys for a deleted user.

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion
\ProfileList\
```

Option 2

1. Edit the relevant file, as follows:
   - Windows Server 2012 R2 and earlier – Edit the EC2Config answer file (`C:\Program Files\Amazon\Ec2ConfigService\sysprep2008.xml`).
   - Windows Server 2016 and 2019 – Edit the unattend.xml answer file (`C:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml`).
   - Windows Server 2022 – Edit the unattend.xml answer file (`C:\ProgramData\Amazon\EC2Launch\sysprep\unattend.xml`).
2. Change `<CopyProfile>true</CopyProfile>` to `<CopyProfile>false</CopyProfile>`.
3. Run Sysprep again. Note that this configuration change will delete the built-in administrator user profile after Sysprep completes.

Troubleshoot impaired Amazon EC2 Linux instance using EC2Rescue

EC2Rescue for Linux is an easy-to-use, open-source tool that can be run on an Amazon EC2 Linux instance to diagnose, troubleshoot, and remediate common issues using its library of over 100 *modules*. Modules are YAML files that contain either a BASH or a Python script and the necessary metadata.

Some generalized use cases for EC2Rescue for Linux instances include:

- Gathering syslog and package manager logs
- Collecting resource utilization data
- Diagnosing and remediating known problematic kernel parameters and common OpenSSH issues

> **Note**
>
> The `AWSSupport-TroubleshootSSH` AWS Systems Manager Automation runbook installs EC2Rescue for Linux and then uses the tool to check or attempt to fix common issues that prevent an SSH connection to a Linux instance. For more information, see AWSSupport-TroubleshootSSH.

If you are using a Windows instance, see the section called "EC2Rescue for Windows instances".

Topics

- Install EC2Rescue on an Amazon EC2 Linux instance
- Run EC2Rescue commands on an Amazon EC2 Linux instance
- Develop EC2Rescue modules for Amazon EC2 Linux instances

Install EC2Rescue on an Amazon EC2 Linux instance

The EC2Rescue for Linux tool can be installed on an Amazon EC2 Linux instance that meets the following prerequisites.

Prerequisites

- Supported operating systems:
  - Amazon Linux 2
  - Amazon Linux 2016.09+
  - SUSE Linux Enterprise Server 12+
  - RHEL 7+
  - Ubuntu 16.04+
- Software requirements:
  - Python 2.7.9+ or 3.2+

Install EC2Rescue

The `AWSSupport-TroubleshootSSH` runbook installs EC2Rescue for Linux and then uses the tool to check or attempt to fix common issues that prevent a remote connection to a Linux machine via SSH. For more information, and to run this automation, see Support-TroubleshootSSH.

If your system has the required Python version, you can install the standard build. Otherwise, you can install the bundled build, which includes a minimal copy of Python.

To install the standard build

1. From a working Linux instance, download the EC2Rescue for Linux tool:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz
```

2. (*Optional*) Verify the signature of the EC2Rescue for Linux installation file. For more information, see (Optional) Verify the signature of EC2Rescue for Linux.
3. Download the sha256 hash file:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sha256
```

4. Verify the integrity of the tarball:

```
sha256sum -c ec2rl.tgz.sha256
```

5. Unpack the tarball:

```
tar -xzvf ec2rl.tgz
```

6. Verify the installation by listing out the help file:

```
cd ec2rl-<version_number>
./ec2rl help
```

To install the bundled build

For a link to the download and a list of limitations, see EC2Rescue for Linux on github.

(Optional) Verify the signature of EC2Rescue for Linux

The following is the recommended process of verifying the validity of the EC2Rescue for Linux package for Linux-based operating systems.

When you download an application from the internet, we recommend that you authenticate the identity of the software publisher and check that the application has not been altered or corrupted after it was published. This protects you from installing a version of the application that contains a virus or other malicious code.

If, after running the steps in this topic, you determine that the software for EC2Rescue for Linux is altered or corrupted, do not run the installation file. Instead, contact Amazon Web Services.

EC2Rescue for Linux files for Linux-based operating systems are signed using GnuPG, an open-source implementation of the Pretty Good Privacy (OpenPGP) standard for secure digital signatures. GnuPG (also known as GPG) provides authentication and integrity checking through a digital signature. AWS publishes a public key and signatures that you can use to verify the downloaded EC2Rescue for Linux package. For more information about PGP and GnuPG (GPG), see https://www.gnupg.org/.

The first step is to establish trust with the software publisher. Download the public key of the software publisher, check that the owner of the public key is who they claim to be, and then add the public key to your keyring. Your keyring is a collection of known public keys. After you establish the authenticity of the public key, you can use it to verify the signature of the application.

Tasks

- Authenticate and import the public key
- Verify the signature of the package

Authenticate and import the public key

The next step in the process is to authenticate the EC2Rescue for Linux public key and add it as a trusted key in your GPG keyring.

To authenticate and import the EC2Rescue for Linux public key

1. At a command prompt, use the following command to obtain a copy of our public GPG build key:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.key
```

2. At a command prompt in the directory where you saved `ec2rl.key`, use the following command to import the EC2Rescue for Linux public key into your keyring:

```
gpg2 --import ec2rl.key
```

The command returns results similar to the following:

```
gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
 imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

> **Tip**
>
> If you see an error stating that the command cannot be found, install the GnuPG utility with `apt-get install gnupg2` (Debian-based Linux) or `yum install gnupg2` (Red Hat- based Linux).

Verify the signature of the package

After you've installed the GPG tools, authenticated and imported the EC2Rescue for Linux public key, and verified that the EC2Rescue for Linux public key is trusted, you are ready to verify the signature of the EC2Rescue for Linux installation script.

To verify the EC2Rescue for Linux installation script signature

1. At a command prompt, run the following command to download the signature file for the installation script:

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sig
```

2. Verify the signature by running the following command at a command prompt in the directory where you saved `ec2rl.tgz.sig` and the EC2Rescue for Linux installation file. Both files must be present.

```
gpg2 --verify ./ec2rl.tgz.sig
```

The output should look something like the following:

```
gpg: Signature made Thu 12 Jul 2018 01:57:51 AM UTC using RSA key ID 6991ED45
gpg: Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E528 BCC9 0DBF 5AFA 0F6C  C36A F780 4843 2FAE 2A1C
     Subkey fingerprint: 966B 0D27 85E9 AEEC 1146  7A9D 8851 1153 6991 ED45
```

If the output contains the phrase `Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"`, it means that the signature has successfully been verified, and you can proceed to run the EC2Rescue for Linux installation script.

If the output includes the phrase `BAD signature`, check whether you performed the procedure correctly. If you continue to get this response, contact Amazon Web Services and do not run the installation file that you downloaded previously.

The following are details about the warnings that you might see:

- **WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner.** This refers to your personal level of trust in your belief that you possess an authentic public key for EC2Rescue for Linux. In an ideal world, you would visit an Amazon Web Services office and receive the key in person. However, more often you download it from a website. In this case, the website is an Amazon Web Services website.
- **gpg2: no ultimately trusted keys found.** This means that the specific key is not "ultimately trusted" by you (or by other people whom you trust).

For more information, see https://www.gnupg.org/.

Run EC2Rescue commands on an Amazon EC2 Linux instance

EC2Rescue is a command line tool. After you have installed EC2Rescue on your Linux instance, you can get general help on how to use the tool by running `./ec2rl help`. You can view the available modules by running `./ec2rl list`, and you can get help on a specific module by running `./ec2rl help` *`module_name`*.

The following are common tasks you can perform to get started using this tool.

Tasks

- Run EC2Rescue modules
- Upload the EC2Rescue module results
- Create backups of an Amazon EC2 Linux instance

Run EC2Rescue modules

To run all EC2Rescue modules

Use the **./ec2rl run** command without specifying any additional parameters. Some modules require root access. If you are not a root user, use **sudo** when you run the command.

```
./ec2rl run
```

To run a specific EC2Rescue module

Use the **./ec2rl run** command and for `--only-modules`, specify the name of the module to run. Some modules require *arguments* to use them.

```
./ec2rl run --only-modules=module_name --arguments
```

For example, to run the **dig** module to query the `amazon.com` domain, use the following command.

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

To view the results of an EC2Rescue module

Run the module then view the log file in `cat /var/tmp/ec2rl/logfile_location`. For example, the log file for the **dig** module can be found in the following location:

```
cat /var/tmp/ec2rl/timestamp/mod_out/run/dig.log
```

Upload the EC2Rescue module results

If Support has requested the results for a EC2Rescue module, you can upload the log file using the EC2Rescue tool. You can upload the results either to a location provided by Support or to an Amazon S3 bucket that you own.

To upload results to a location provided by Support

Use the **./ec2rl upload** command. For `--upload-directory`, specify the location of the log file. For `--support-url`, specify the URL provided by Support.

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --support-
url="url_provided_by_aws_support"
```

To upload results to an Amazon S3 bucket

Use the **./ec2rl upload** command. For `--upload-directory`, specify the location of the log file. For `--presigned-url`, specify a presigned URL for the S3 bucket. For more information about generating pre-signed URLs for Amazon S3, see Uploading Objects Using Pre-Signed URLs.

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/logfile_location --presigned-
url="presigned_s3_url"
```

Create backups of an Amazon EC2 Linux instance

You can use EC2Rescue to backup your Linux instance by creating an AMI or by creating snapshots of it's attached volumes.

To create an AMI

Use the `./ec2rl run` command and for --backup, specify `ami`.

```
./ec2rl run --backup=ami
```

To create multi-volume snapshots of all attached volumes

Use the `./ec2rl run` command and for --backup, specify `allvolumes`.

```
./ec2rl run --backup=allvolumes
```

To create a snapshot of a specific attached volume

Use the `./ec2rl run` command and for `--backup`, specify the ID of the volume to back up.

```
./ec2rl run --backup=vol-01234567890abcdef
```

Develop EC2Rescue modules for Amazon EC2 Linux instances

Modules are written in YAML, a data serialization standard. A module's YAML file consists of a single document, representing the module and its attributes.

Add module attributes

The following table lists the available module attributes.

Attribute	Description
name	The name of the module. The name should be less than or equal to 18 characters in length.
version	The version number of the module.
title	A short, descriptive title for the module. This value should be less than or equal to 50 characters in length.

helptext	The extended description of the module. Each line should be less than or equal to 75 characters in length. If the module consumes arguments, required or optional, include them in the helptext value.
	
	For example:
	
	`helptext: !!str |`
	`  Collect output from ps for system analysis`
	`  Consumes --times= for number of times to repeat`
	`  Consumes --period= for time period between repetition`

Attribute	Description
placement	The stage in which the module should be run. Supported values: • prediagnostic • run • postdiagnostic
language	The language that the module code is written in. Supported values: • bash • python **Note** Python code must be compatible with both Python 2.7.9+ and Python 3.2+.
remediation	Indicates whether the module supports remediation. Supported values are `True` or `False`. The module defaults to `False` if this is absent, making it an optional attribute for those modules that do not support remediation.
content	The entirety of the script code.
constraint	The name of the object containing the constraint values.

Attribute	Description
domain	A descriptor of how the module is grouped or classified. The set of included modules uses the following domains: • application • net • os • performance
class	A descriptor of the type of task performed by the module. The set of included modules uses the following classes: • collect (collects output from programs) • diagnose (pass/fail based on a set of criteria) • gather (copies files and writes to specific file)
distro	The list of Linux distributions that this module supports. The set of included modules uses the following distributions: • **alami** (Amazon Linux) • **rhel** • **ubuntu** • **suse**
required	The required arguments that the module is consuming from the CLI options.
optional	The optional arguments that the module can use.

Attribute	Description
software	The software executables used in the module. This attribute is intended to specify software that is not installed by default. The EC2Rescue for Linux logic ensures that these programs are present and executable before running the module.
package	The source software package for an executable. This attribute is intended to provide extended details on the package with the software, including a URL for downloading or getting further information.
sudo	Indicates whether root access is required to run the module. You do not need to implement sudo checks in the module script. If the value is true, then the EC2Rescue for Linux logic only runs the module when the executing user has root access.
perfimpact	Indicates whether the module can have significant performance impact upon the environment in which it is run. If the value is true and the `--perfimpact=true` argument is not present, then the module is skipped.
parallelexclusive	Specifies a program that requires mutual exclusivity. For example, all modules specifying "bpf" run in a serial manner.

Add environment variables

The following table lists the available environment variables.

Environment Variable	Description
EC2RL_CALLPATH	The path to `ec2rl.py`. This path can be used to locate the lib directory and use vendored Python modules.
EC2RL_WORKDIR	The main tmp directory for the diagnostic tool. Default value: `/var/tmp/ec2rl` .
EC2RL_RUNDIR	The directory where all output is stored. Default value: `/var/tmp/ec2rl/<date&timestamp>` .
EC2RL_GATHEREDDIR	The root directory for placing gathered module data. Default value:`/var/tmp/ec2rl/<date&timestamp>/mod_out/gathered/` .
EC2RL_NET_DRIVER	The driver in use for the first, alphabetically ordered, non-virtual network interface on the instance. Examples: • xen_netfront • ixgbevf • ena
EC2RL_SUDO	True if EC2Rescue for Linux is running as root; otherwise, false.
EC2RL_VIRT_TYPE	The virtualization type as provided by the instance metadata. Examples:

Environment Variable	Description
	• default-hvm • default-paravirtual
EC2RL_INTERFACES	An enumerated list of interfaces on the system. The value is a string containing names, such as `eth0`, `eth1`, etc. This is generated via the `functions.bash` and is only available for modules that have sourced it.

Use YAML syntax

The following should be noted when constructing your module YAML files:

- The triple hyphen (`---`) denotes the explicit start of a document.
- The `!ec2rlcore.module.Module` tag tells the YAML parser which constructor to call when creating the object from the data stream. You can find the constructor inside the `module.py` file.
- The `!!str` tag tells the YAML parser to not attempt to determine the type of data, and instead interpret the content as a string literal.
- The pipe character (`|`) tells the YAML parser that the value is a literal-style scalar. In this case, the parser includes all whitespace. This is important for modules because indentation and newline characters are kept.
- The YAML standard indent is two spaces, which can be seen in the following examples. Ensure that you maintain standard indentation (for example, four spaces for Python) for your script and then indent the entire content two spaces inside the module file.

Example modules

Example one (`mod.d/ps.yaml`):

```
--- !ec2rlcore.module.Module
# Module document. Translates directly into an almost-complete Module object
name: !!str ps
path: !!str
```

```
version: !!str 1.0
title: !!str Collect output from ps for system analysis
helptext: !!str |
  Collect output from ps for system analysis
  Requires --times= for number of times to repeat
  Requires --period= for time period between repetition
placement: !!str run
package:
  - !!str
language: !!str bash
content: !!str |
  #!/bin/bash
  error_trap()
  {
      printf "%0.s=" {1..80}
      echo -e "\nERROR: "$BASH_COMMAND" exited with an error on line ${BASH_LINENO[0]}"
      exit 0
  }
  trap error_trap ERR

  # read-in shared function
  source functions.bash
  echo "I will collect ps output from this $EC2RL_DISTRO box for $times times every
 $period seconds."
  for i in $(seq 1 $times); do
      ps auxww
      sleep $period
  done
constraint:
  requires_ec2: !!str False
  domain: !!str performance
  class: !!str collect
  distro: !!str alami ubuntu rhel suse
  required: !!str period times
  optional: !!str
  software: !!str
  sudo: !!str False
  perfimpact: !!str False
  parallelexclusive: !!str
```

Troubleshoot impaired Amazon EC2 Windows instance using EC2Rescue

EC2Rescue for Windows Server is an easy-to-use tool that you run on an Amazon EC2 Windows Server instance to diagnose and troubleshoot possible problems. It is valuable for collecting log files and troubleshooting issues and also proactively searching for possible areas of concern. It can even examine Amazon EBS root volumes from other instances and collect relevant logs for troubleshooting Windows Server instances using that volume. The following are some common issues that EC2Rescue can address:

- Instance connectivity issues due to firewall, Remote Desktop Protocol (RDP), or network interface configuration
- Operating system boot issues due to a stop error, boot loop, or corrupted registry
- Issues that might need advanced log analysis and troubleshooting

EC2Rescue for Windows Server has two different modules:

- A **data collector module** that collects data from all different sources
- An **analyzer module** that parses the data collected against a series of predefined rules to identify issues and provide suggestions

The EC2Rescue for Windows Server tool only runs on Amazon EC2 instances running Windows Server 2012 and later. When the tool starts, it checks whether it is running on an Amazon EC2 instance.

> **Note**
>
> The `AWSSupport-ExecuteEC2Rescue` AWS Systems Manager Automation runbook uses the EC2Rescue tool to troubleshoot and, where possible, fix common connectivity issues with the specified EC2 instance. For more information, and to run this automation, see >AWSSupport-ExecuteEC2Rescue.

If you are using a Linux instance, see the section called “EC2Rescue for Linux instances”.

Topics

Troubleshoot impaired Windows instance with the EC2Rescue GUI

EC2Rescue for Windows Server can perform the following analysis on **offline instances**:

Option	Description

Diagnose and Rescue	EC2Rescue for Windows Server can detect and address issues with the following service settings:
	• System Time
	  • **RealTimeisUniversal** - Detects whether the `RealTimeisUniversal` registry key is enabled. If disabled, Windows system time drifts when the timezone is set to a value other than UTC.
	• Windows Firewall
	  • **Domain networks** - Detects whether this Windows Firewall profile is enabled or disabled.
	  • **Private networks** - Detects whether this Windows Firewall profile is enabled or disabled.
	  • **Guest or public networks** - Detects whether this Windows Firewall profile is enabled or disabled.
	• Remote Desktop
	  • **Service Start** - Detects whether the Remote Desktop service is enabled.

Option	Description

	• **Remote Desktop Connections** - Detects whether this is enabled.
	• **TCP Port** - Detects which port the Remote Desktop service is listening on.
	
	• EC2Config (Windows Server 2012 R2 and earlier)
	  • **Installation** - Detects which EC2Config version is installed.
	  • **Service Start** - Detects whether the EC2Config service is enabled.
	  • **Ec2SetPassword** - Generates a new administrator password.
	  • **Ec2HandleUserData** - Allows you to run a user data script on the next boot of the instance.
	
	• EC2Launch (Windows Server 2016 and later)
	  • **Installation** - Detects which EC2Launch version is installed.
	  • **Ec2SetPassword** - Generates a new administrator password.
	
	• Network Interface
	  • **DHCP Service Startup** - Detects whether the DHCP service is enabled.
	  • **Ethernet detail** - Displays informati on about the network driver version, if detected.
	  • **DHCP on Ethernet** - Detects whether DHCP is enabled.

Option	Description
	• Disk signature status • **Signature on disk** and **Signature on Boot Configuration Database (BCD)** - Detects whether the disk signature and the BCD signature are the same. If the values are different, EC2Rescue attempts to overwrite the disk signature with the signature on BCD.
Restore	Perform one of the following actions: • **Last Known Good Configuration** - Attempts to boot the instance into the last known bootable state. • **Restore registry from backup** - Restores the registry from `\Windows\System32\config\RegBack` .
Capture Logs	Allows you to capture logs on the instance for analysis.

EC2Rescue for Windows Server can collect the following data from **active and offline instances**:

Item	Description
Event Log	Collects application, system, and EC2Config event logs.
Registry	Collects `SYSTEM` and `SOFTWARE` hives.
Windows Update Log	Collects log files generated by Windows Update.

Item	Description
	Note In Windows Server 2016 and later, the log is collected in Event Tracing for Windows (ETW) format.
Sysprep Log	Collects log files generated by the Windows System Preparation tool.
Driver Setup Log	Collects Windows SetupAPI logs (`setupapi.dev.log` and `setupapi.setup.log`).
Boot Configuration	Collects `HKEY_LOCAL_MACHINE\BCD00000000` hive.
Memory Dump	Collects any memory dump files that exist on the instance.
EC2Config File	Collects log files generated by the EC2Config service.
EC2Launch File	Collects log files generated by the EC2Launch scripts.
SSM Agent File	Collects log files generated by SSM Agent and Patch Manager logs.
EC2 ElasticGPUs File	Collects event logs related to elastic GPUs.
ECS	Collects logs related to Amazon ECS.
CloudEndure	Collects log files related to CloudEndure Agent.
AWS Replication Agent for MGN or DRS Log Files	Collects log files related to AWS Application Migration Service or AWS Elastic Disaster Recovery.

EC2Rescue for Windows Server can collect the following additional data from **active instances**:

Item	Description
System Information	Collects MSInfo32.
Group Policy Result	Collects a Group Policy report.

Analyze an offline instance

The **Offline Instance** option is useful for debugging boot issues with Windows instances.

To perform an action on an offline instance

1. From a working Windows Server instance, download the EC2Rescue for Windows Server tool and extract the files.

   You can run the following PowerShell command to download EC2Rescue without changing your Internet Explorer Enhanced Security Configuration (ESC):

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   This command will download the EC2Rescue .zip file to the desktop of the currently logged in user.

   > **Note**
   >
   > If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:
   >
   > ```
   > [Net.ServicePointManager]::SecurityProtocol =
   >  [Net.SecurityProtocolType]::Tls12
   > ```

2. Stop the faulty instance, if it is not stopped already.

3. Detach the EBS root volume from the faulty instance and attach the volume to a working Windows instance that has EC2Rescue for Windows Server installed.
4. Run the EC2Rescue for Windows Server tool on the working instance and choose **Offline Instance**.
5. Select the disk of the newly mounted volume and choose **Next**.
6. Confirm the disk selection and choose **Yes**.
7. Choose the offline instance option to perform and choose **Next**.

The EC2Rescue for Windows Server tool scans the volume and collects troubleshooting information based on the selected log files.

Collect data from an active instance

You can collect logs and other data from an active instance.

To collect data from an active instance

1. Connect to your Windows instance.
2. Download the EC2Rescue for Windows Server tool to your Windows instance and extract the files.

   You can run the following PowerShell command to download EC2Rescue without changing your Internet Explorer Enhanced Security Configuration (ESC):

   ```
   Invoke-WebRequest https://s3.amazonaws.com/ec2rescue/windows/EC2Rescue_latest.zip -
   OutFile $env:USERPROFILE\Desktop\EC2Rescue_latest.zip
   ```

   This command will download the EC2Rescue .zip file to the desktop of the currently logged in user.

   > **Note**
   >
   > If you receive an error when downloading the file, and you are using Windows Server 2016 or earlier, TLS 1.2 might need to be enabled for your PowerShell terminal. You can enable TLS 1.2 for the current PowerShell session with the following command and then try again:

```
[Net.ServicePointManager]::SecurityProtocol =
 [Net.SecurityProtocolType]::Tls12
```

3. Open the EC2Rescue for Windows Server application and accept the license agreement.
4. Choose **Next**, **Current instance**, **Capture logs**.
5. Select the data items to collect and choose **Collect...**. Read the warning and choose **Yes** to continue.
6. Choose a file name and location for the ZIP file and choose **Save**.
7. After EC2Rescue for Windows Server completes, choose **Open Containing Folder** to view the ZIP file.
8. Choose **Finish**.

Troubleshoot impaired Windows instance with the EC2Rescue CLI

The EC2Rescue for Windows Server command line interface (CLI) allows you to run an EC2Rescue for Windows Server plugin (referred as an "action") programmatically.

The EC2Rescue for Windows Server tool has two execution modes:

- **/online**—This allows you to take action on the instance that EC2Rescue for Windows Server is installed on, such as collect log files.
- **/offline:<device_id>**—This allows you to take action on the offline root volume that is attached to a separate Amazon EC2 Windows instance, on which you have installed EC2Rescue for Windows Server.

Download the EC2Rescue for Windows Server tool to your Windows instance and extract the files. You can view the help file using the following command:

```
EC2RescueCmd.exe /help
```

EC2Rescue for Windows Server can perform the following actions on an Amazon EC2 Windows instance:

- Collect action
- Rescue action

- Restore action

Collect action

> **Note**
>
> You can collect all logs, an entire log group, or an individual log within a group.

EC2Rescue for Windows Server can collect the following data from **active and offline instances**.

Log group	Available logs	Description
`all`		Collects all available logs.
`eventlog`	• `'Application'` • `'System'` • `'EC2ConfigService'`	Collects application, system, and EC2Config event logs.
`memory-dump`	• `'Memory Dump File'` • `'Mini Dump Files'`	Collects any memory dump files that exist on the instance.
`ec2config`	• `'Log Files'` • `'Configuration Files'`	Collects log files generated by the EC2Config service.
`ec2launch`	• `'Logs'` • `'Config'`	Collects log files generated by the EC2Launch scripts.
`ssm-agent`	• `'Log Files'` • `'Patch Baseline Logs'` • `'InstanceData'`	Collects log files generated by SSM Agent and Patch Manager logs.
`sysprep`	`'Log Files'`	Collects log files generated by the Windows System Preparation tool.

Log group	Available logs	Description
`driver-setup`	• `'SetupAPI Log Files'` • `'DPInst Log File'` • `'AWS PV Setup Log File'`	Collects Windows SetupAPI logs (`setupapi.dev.log` and `setupapi.setup.log`).
`registry`	• `'SYSTEM'` • `'SOFTWARE'` • `'BCD'`	Collects `SYSTEM` and `SOFTWARE` hives.
`egpu`	• `'Event Log'` • `'System Files'`	Collects event logs related to elastic GPUs.
`boot-config`	`'BCDEDIT Output'`	Collects `HKEY_LOCAL_MACHINE\BCD00000000` hive.
`windows-update`	`'Log Files'`	Collects log files generated by Windows Update. **Note** In Windows Server 2016 and later, the log is collected in Event Tracing for Windows (ETW) format.
`cloudendure`	• `'Migrate Script Logs'` • `'Driver Logs'` • `'CloudEndure File List'`	Collects log files related to CloudEndure Agent.

EC2Rescue for Windows Server can collect the following additional data from **active instances**.

Log group	Available logs	Description
`system-info`	`'MSInfo32 Output'`	Collects `MSInfo32`.
`gpresult`	`'GPResult Output'`	Collects a Group Policy report.

The following are the available options:

- **/output:<outputFilePath>** - Required destination file path location to save collected log files in zip format.
- **/no-offline** - Optional attribute used in offline mode. Does not set the volume offline after completing the action.
- **/no-fix-signature** - Optional attribute used in offline mode. Does not fix a possible disk signature collision after completing the action.

Examples

The following are examples using the EC2Rescue for Windows Server CLI.

Online mode examples

Collect all available logs:

```
EC2RescueCmd /accepteula /online /collect:all /output:<outputFilePath>
```

Collect only a specific log group:

```
EC2RescueCmd /accepteula /online /collect:ec2config /output:<outputFilePath>
```

Collect individual logs within a log group:

```
EC2RescueCmd /accepteula /online /collect:'ec2config.Log Files,driver-setup.SetupAPI
 Log Files' /output:<outputFilePath>
```

Offline mode examples

Collect all available logs from an EBS volume. The volume is specified by the device_id value.

```
EC2RescueCmd /accepteula /offline:xvdf /collect:all /output:<outputFilePath>
```

Collect only a specific log group:

```
EC2RescueCmd /accepteula /offline:xvdf /collect:ec2config /output:<outputFilePath>
```

Rescue action

EC2Rescue for Windows Server can detect and address issues with the following service settings:

Service group	Available actions	Description
`all`		
`system-time`	`'RealTimeIsUniversal'`	System Time • **RealTimeisUniversal** - Detects whether the `RealTimeisUniversal` registry key is enabled. If disabled, Windows system time drifts when the timezone is set to a value other than UTC.
`firewall`	• `'Domain networks'` • `'Private networks'` • `'Guest or public networks'`	Windows Firewall • **Domain networks** - Detects whether this Windows Firewall profile is enabled or disabled. • **Private networks** - Detects whether this Windows Firewall profile is enabled or disabled.

Service group	Available actions	Description
		• **Guest or public networks** - Detects whether this Windows Firewall profile is enabled or disabled.
rdp	• `'Service Start'` • `'Remote Desktop Connections'` • `'TCP Port'`	Remote Desktop • **Service Start** - Detects whether the Remote Desktop service is enabled. • **Remote Desktop Connections** - Detects whether this is enabled. • **TCP Port** - Detects which port the Remote Desktop service is listening on.
`ec2config`	• `'Service Start'` • `'Ec2SetPassword'` • `'Ec2HandleUserData'`	EC2Config • **Service Start** - Detects whether the EC2Config service is enabled. • **Ec2SetPassword** - Generates a new administrator password. • **Ec2HandleUserData** - Allows you to run a user data script on the next boot of the instance.
`ec2launch`	`'Reset Administrator Password'`	Generates a new Windows administrator password.

Service group	Available actions	Description
`network`	`'DHCP Service Startup'`	Network Interface • **DHCP Service Startup** - Detects whether the DHCP service is enabled.

The following are the available options:

- **/level:<level>** - Optional attribute for the check level that the action should trigger. Allowed values are: `information`, `warning`, `error`, `all`. By default, it is set to `error`.
- **/check-only** - Optional attribute that generates a report but makes no modifications to the offline volume.

> **Note**
>
> If EC2Rescue for Windows Server detects a possible disk signature collision, it corrects the signature after the offline process completes by default, even when you use the `/check-only` option. You must use the `/no-fix-signature` option to prevent the correction.

- **/no-offline** - Optional attribute that prevents the volume from being set offline after completing the action.
- **/no-fix-signature** - Optional attribute that does not fix a possible disk signature collision after completing the action.

Rescue examples

The following are examples using the EC2Rescue for Windows Server CLI. The volume is specified using the device_id value.

Attempt to fix all identified issues on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:all
```

Attempt to fix all issues within a service group on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:firewall
```

Attempt to fix a specific item within a service group on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:rdp.'Service Start'
```

Specify multiple issues to attempt to fix on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /rescue:'system-
time.RealTimeIsUniversal,ec2config.Service Start'
```

Restore action

EC2Rescue for Windows Server can detect and address issues with the following service settings:

Service Group	Available Actions	Description
Restore Last Known Good Configuration	`lkgc`	**Last Known Good Configuration** - Attempts to boot the instance into the last known bootable state.
Restore Windows registry from latest backup	`regback`	**Restore registry from backup** - Restores the registry from `\Windows\System32\config\RegBack .`

The following are the available options:

- **/no-offline**—Optional attribute that prevents the volume from being set offline after completing the action.
- **/no-fix-signature**—Optional attribute that does not fix a possible disk signature collision after completing the action.

Restore examples

The following are examples using the EC2Rescue for Windows Server CLI. The volume is specified using the device_id value.

Restore last known good configuration on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:lkgc
```

Restore the last Windows registry backup on a volume:

```
EC2RescueCmd /accepteula /offline:xvdf /restore:regback
```

Troubleshoot impaired Windows instance with EC2Rescue and Systems Manager

Support provides you with a Systems Manager Run Command document to interface with your Systems Manager-enabled instance to run EC2Rescue for Windows Server. The Run Command document is called `AWSSupport-RunEC2RescueForWindowsTool`.

This Systems Manager Run Command document performs the following tasks:

- Downloads and verifies EC2Rescue for Windows Server.
- Imports a PowerShell module to ease your interaction with the tool.
- Runs EC2RescueCmd with the provided command and parameters.

The Systems Manager Run Command document accepts three parameters:

- **Command**—The EC2Rescue for Windows Server action. The current allowed values are:
  - **ResetAccess**—Resets the local Administrator password. The local Administrator password of the current instance will be reset and the randomly generated password will be securely stored in Parameter Store as `/EC2Rescue/Password/<INSTANCE_ID>`. If you select this action and provide no parameters, passwords are encrypted automatically with the default KMS key. Optionally, you can specify a KMS key ID in Parameters to encrypt the password with your own key.
  - **CollectLogs**—Runs EC2Rescue for Windows Server with the `/collect:all` action. If you select this action, `Parameters` must include an Amazon S3 bucket name to upload the logs to.

- **FixAll**—Runs EC2Rescue for Windows Server with the `/rescue:all` action. If you select this action, `Parameters` must include the block device name to rescue.
- **Parameters**—The PowerShell parameters to pass for the specified command.

Requirements

To run the **ResetAccess** action, your Amazon EC2 instance must have the a policy attached that grants permissions to write the encrypted password to Parameter Store. After attaching the policy, wait a few minutes before attempting to reset the password of an instance after you have attached this policy to the related IAM role.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    }
  ]
}
```

If you are using a custom KMS key, not the default KMS key, use this policy instead.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "ssm:PutParameter"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/EC2Rescue/
Passwords/<instanceid>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/<kmskeyid>"
      ]
    }
  ]
}
```

View the JSON for the document

The following procedure describes how to view the JSON for this document.

To view the JSON for the Systems Manager Run Command document

1. Open the AWS Systems Manager console at https://console.aws.amazon.com/systems-manager/.
2. In the navigation pane, expand **Change Management Tools** and choose **Documents**.
3. In the search bar, enter `AWSSupport-RunEC2RescueForWindowsTool`, and then select the `AWSSupport-RunEC2RescueForWindowsTool` document.
4. Choose the **Content** tab.

Examples

Here are some examples on how to use the Systems Manager Run Command document to run EC2Rescue for Windows Server, using the AWS CLI. For more information about sending commands using the AWS CLI, see send-command.

Examples

- Attempt to fix all identified issues on an offline root volume
- Collect logs from the current Amazon EC2 Windows instance
- Reset the local Administrator password

Attempt to fix all identified issues on an offline root volume

Attempt to fix all identified issues on an offline root volume attached to an Amazon EC2 Windows instance:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=FixAll, Parameters='xvdf'" --output
 text
```

Collect logs from the current Amazon EC2 Windows instance

Collect all logs from the current online Amazon EC2 Windows instance and upload them to an Amazon S3 bucket:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=CollectLogs, Parameters='amzn-s3-
demo-bucket'" --output text
```

Reset the local Administrator password

The following examples show methods you can use to reset the local Administrator password. The output provides a link to Parameter Store, where you can find the randomly generated secure password you can then use to RDP to your Amazon EC2 Windows instance as the local Administrator.

Reset the local Administrator password of an online instance using the default AWS KMS key alias/aws/ssm:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess" --output text
```

Reset the local Administrator password of an online instance using a KMS key:

```
aws ssm send-command --instance-ids "i-0cb2b964d3e14fd9f" --document-name "AWSSupport-
RunEC2RescueForWindowsTool" --parameters "Command=ResetAccess, Parameters=a133dc3c-
a2g4-4fc6-a873-6c0720104bf0" --output text
```

Note

In this example, the KMS key is `a133dc3c-a2g4-4fc6-a873-6c0720104bf0`.

EC2 Serial Console for instances

With the EC2 serial console, you have access to your Amazon EC2 instance's serial port, which you can use to troubleshoot boot, network configuration, and other issues. The serial console does not require your instance to have any networking capabilities. With the serial console, you can enter commands to an instance as if your keyboard and monitor are directly attached to the instance's serial port. The serial console session lasts during instance reboot and stop. During reboot, you can view all of the boot messages from the start.

Access to the serial console is not available by default. Your organization must grant account access to the serial console and configure IAM policies to grant your users access to the serial console. Serial console access can be controlled at a granular level by using instance IDs, resource tags, and other IAM levers. For more information, see Configure access to the EC2 Serial Console.

The serial console can be accessed by using the EC2 console or the AWS CLI.

The serial console is available at no additional cost.

Topics

Prerequisites for the EC2 Serial Console

To connect to the EC2 Serial Console and use your chosen tool for troubleshooting, the following prerequisites must be met:

- AWS Regions
- Wavelength Zones and AWS Outposts
- Local Zones
- Instance types
- Grant access
- Support for browser-based client
- Instance state
- Amazon EC2 Systems Manager
- Configure your chosen troubleshooting tool

AWS Regions

Supported in all AWS Regions except Asia Pacific (Taipei).

Wavelength Zones and AWS Outposts

Not supported.

Local Zones

Supported in all Local Zones.

Instance types

Supported instance types:

- **Linux**
  - All virtualized instances built on the Nitro System.
  - All bare metal instances except:
    - General purpose: `a1.metal`, `mac1.metal`, `mac2.metal`
    - Accelerated computing: `g5g.metal`

  - Memory optimized: `u-6tb1.metal`, `u-9tb1.metal`, `u-12tb1.metal`, `u-18tb1.metal`, `u-24tb1.metal`
- **Windows**

  All virtualized instances built on the Nitro System. Not supported on bare metal instances.

Grant access

You must complete the configuration tasks to grant access to the EC2 Serial Console. For more information, see Configure access to the EC2 Serial Console.

Support for browser-based client

To connect to the serial console using the browser-based client, your browser must support WebSocket. If your browser does not support WebSocket, connect to the serial console using your own key and an SSH client.

Instance state

Must be `running`.

You can't connect to the serial console if the instance is in the `pending`, `stopping`, `stopped`, `shutting-down`, or `terminated` state.

For more information about the instance states, see Amazon EC2 instance state changes.

Amazon EC2 Systems Manager

If the instance uses Amazon EC2 Systems Manager, then SSM Agent version 3.0.854.0 or later must be installed on the instance. For information about SSM Agent, see Working with SSM Agent in the *AWS Systems Manager User Guide*.

Configure your chosen troubleshooting tool

To troubleshoot your instance using the serial console, you can use GRUB or SysRq on Linux instances, and Special Admin Console (SAC) on Windows instances. Before you can use these tools, you must first perform configuration steps on every instance on which you'll use them.

Use the instructions for your instance's operating system to configure your chosen troubleshooting tool.

(Linux instances) Configure GRUB

To configure GRUB, choose one of the following procedures based on the AMI that was used to launch the instance.

Amazon Linux 2

To configure GRUB on an Amazon Linux 2 instance

1. Connect to your Linux instance using SSH
2. Add or change the following options in `/etc/default/grub`:

   - Set `GRUB_TIMEOUT=1`.
   - Add `GRUB_TERMINAL="console serial"`.
   - Add `GRUB_SERIAL_COMMAND="serial --speed=115200"`.

   The following is an example of `/etc/default/grub`. You might need to change the configuration based on your system setup.

   ```
   GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
    biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.shell=0"
   GRUB_TIMEOUT=1
   GRUB_DISABLE_RECOVERY="true"
   GRUB_TERMINAL="console serial"
   GRUB_SERIAL_COMMAND="serial --speed=115200"
   ```

3. Apply the updated configuration by running the following command.

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

Ubuntu

To configure GRUB on an Ubuntu instance

1. Connect to your instance.
2. Add or change the following options in `/etc/default/grub.d/50-cloudimg-settings.cfg`:

   - Set `GRUB_TIMEOUT=1`.

- Add GRUB_TIMEOUT_STYLE=menu.
- Add GRUB_TERMINAL="console serial".
- Remove GRUB_HIDDEN_TIMEOUT.
- Add GRUB_SERIAL_COMMAND="serial --speed=115200".

The following is an example of /etc/default/grub.d/50-cloudimg-settings.cfg. You might need to change the configuration based on your system setup.

```
# Cloud Image specific Grub settings for Generic Cloud Images
# CLOUD_IMG: This file was created/modified by the Cloud Image build process

# Set the recordfail timeout
GRUB_RECORDFAIL_TIMEOUT=0

# Do not wait on grub prompt
GRUB_TIMEOUT=1
GRUB_TIMEOUT_STYLE=menu

# Set the default commandline
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0
 nvme_core.io_timeout=4294967295"

# Set the grub console type
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed 115200"
```

3. Apply the updated configuration by running the following command.

```
[ec2-user ~]$ sudo update-grub
```

RHEL

To configure GRUB on a RHEL instance

1. Connect to your instance.
2. Add or change the following options in /etc/default/grub:
   - Remove GRUB_TERMINAL_OUTPUT.

- Add `GRUB_TERMINAL="console serial"`.
- Add `GRUB_SERIAL_COMMAND="serial --speed=115200"`.

The following is an example of `/etc/default/grub`. You might need to change the configuration based on your system setup.

```
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 rd.blacklist=nouveau nvme_core.io_timeout=4294967295 crashkernel=auto"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
GRUB_TERMINAL="console serial"
GRUB_SERIAL_COMMAND="serial --speed=115200"
```

3. Apply the updated configuration by running the following command.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg --update-bls-cmdline
```

For RHEL 9.2 and earlier, use the following command.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

CentOS

For instances that are launched using a CentOS AMI, GRUB is configured for the serial console by default.

The following is an example of `/etc/default/grub`. Your configuration might be different based on your system setup.

```
GRUB_TIMEOUT=1
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL="serial console"
GRUB_SERIAL_COMMAND="serial --speed=115200"
```

```
GRUB_CMDLINE_LINUX="console=tty0 crashkernel=auto console=ttyS0,115200"
GRUB_DISABLE_RECOVERY="true"
```

(Linux instances) Configure SysRq

To configure SysRq, you enable the SysRq commands for the current boot cycle. To make the configuration persistent, you can also enable the SysRq commands for subsequent boots.

To enable all SysRq commands for the current boot cycle

1. Connect to your instance.
2. Run the following command.

   ```
   [ec2-user ~]$ sudo sysctl -w kernel.sysrq=1
   ```

   This setting will clear on the next reboot.

To enable all SysRq commands for subsequent boots

1. Create the file `/etc/sysctl.d/99-sysrq.conf` and open it in your favorite editor.

   ```
   [ec2-user ~]$ sudo vi /etc/sysctl.d/99-sysrq.conf
   ```

2. Add the following line.

   ```
   kernel.sysrq=1
   ```

3. Reboot the instance to apply the changes.

   ```
   [ec2-user ~]$ sudo reboot
   ```

4. At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
5. At the `Password` prompt, enter the password, and then press **Enter**.

(Windows instances) Enable SAC and the boot menu

> **Note**
>
> If you enable SAC on an instance, the EC2 services that rely on password retrieval will not work from the Amazon EC2 console. Windows on Amazon EC2 launch agents (EC2Config, EC2Launch v1, and EC2Launch v2) rely on the serial console to execute various tasks. These tasks do not perform successfully when you enable SAC on an instance. For more information about Windows on Amazon EC2 launch agents, see the section called "Configure Windows instances". If you enable SAC, you can disable it later. For more information, see Disable SAC and the boot menu.

Use one of the following methods to enable SAC and the boot menu on an instance.

PowerShell

To enable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from an elevated PowerShell command line.
2. Enable SAC.

```
bcdedit /ems '{current}' on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. Enable the boot menu.

```
bcdedit /set '{bootmgr}' displaybootmenu yes
bcdedit /set '{bootmgr}' timeout 15
bcdedit /set '{bootmgr}' bootems yes
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Command prompt

To enable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from the command prompt.
2. Enable SAC.

```
bcdedit /ems {current} on
bcdedit /emssettings EMSPORT:1 EMSBAUDRATE:115200
```

3. Enable the boot menu.

```
bcdedit /set {bootmgr} displaybootmenu yes
bcdedit /set {bootmgr} timeout 15
bcdedit /set {bootmgr} bootems yes
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Configure access to the EC2 Serial Console

To configure access to the serial console, you must grant serial console access at the account level and then configure IAM policies to grant access to your users. For Linux instances you must also configure a password-based user on every instance so that your users can use the serial console for troubleshooting.

EC2 Serial Console uses a virtual serial port connection to interact with an instance. This connection is independent of the instance VPC, so that you can work with the instance operating system and run troubleshooting tools even if there are boot failures or network configuration issues. Because this connection is outside of the VPC network, EC2 Serial Console does not use the instance security group or the subnet network ACL to authorize traffic to the instance.

Before you begin

Verify that the prerequisites are met.

Contents

- Manage account access to the EC2 Serial Console
- Configure IAM policies for EC2 Serial Console access
- Set an OS user password on a Linux instance

Levels of access to the EC2 Serial Console

By default, there is no access to the serial console at the account level. You need to explicitly grant access to the serial console at the account level. For more information, see Manage account access to the EC2 Serial Console.

You can use a service control policy (SCP) to allow access to the serial console within your organization. You can then have granular access control at the user level by using an IAM policy to control access. By using a combination of SCP and IAM policies, you have different levels of access control to the serial console.

Organization level

You can use a service control policy (SCP) to allow access to the serial console for member accounts within your organization. For more information about SCPs, see Service control policies in the *AWS Organizations User Guide*.

Instance level

You can configure the serial console access policies by using IAM PrincipalTag and ResourceTag constructions and by specifying instances by their ID. For more information, see Configure IAM policies for EC2 Serial Console access.

User level

You can configure access at the user level by configuring an IAM policy to allow or deny a specified user the permission to push the SSH public key to the serial console service of a particular instance. For more information, see Configure IAM policies for EC2 Serial Console access.

OS level (Linux instances only)

You can set a user password at the guest OS level. This provides access to the serial console for some use cases. However, to monitor the logs, you don't need a password-based user. For more information, see Set an OS user password on a Linux instance.

Manage account access to the EC2 Serial Console

By default, there is no access to the serial console at the account level. You need to explicitly grant access to the serial console at the account level.

This setting is configured at the account level, either directly in the account or by using a declarative policy. It must be configured in each AWS Region where you want to grant access to the serial console. Using a declarative policy allows you to apply the setting across multiple Regions simultaneously, as well as across multiple accounts simultaneously. When a declarative policy is in use, you can't modify the setting directly within an account. This topic describes how to configure the setting directly within an account. For information about using declarative policies, see Declarative policies in the *AWS Organizations User Guide.*

Contents

Grant permission to users to manage account access

To allow your users to manage account access to the EC2 serial console, you need to grant them the required IAM permissions.

The following policy grants permissions to view the account status, and to allow and prevent account access to the EC2 serial console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetSerialConsoleAccessStatus",
                "ec2:EnableSerialConsoleAccess",
                "ec2:DisableSerialConsoleAccess"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

For more information, see Creating IAM policies in the *IAM User Guide*.

View account access status to the serial console

Console

To view account access to the serial console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.
4. On the **EC2 Serial Console** tab, the value of **EC2 Serial Console access** is either **Allowed** or **Prevented**.

AWS CLI

To view account access to the serial console

Use the get-serial-console-access-status command.

```
aws ec2 get-serial-console-access-status
```

The following is example output. A value of `true` indicates that the account is allowed access to the serial console.

```
{
    "SerialConsoleAccessEnabled": true,
    "ManagedBy": "account"
}
```

The `ManagedBy` field indicates the entity that configured the setting. The possible values are `account` (configured directly) or `declarative-policy`. For more information, see Declarative policies in the *AWS Organizations User Guide*.

PowerShell

To view account access to the serial console

Use the Get-EC2SerialConsoleAccessStatus cmdlet.

```
Get-EC2SerialConsoleAccessStatus -Select *
```

The following is example output. A value of `True` indicates that the account is allowed access to the serial console.

```
ManagedBy SerialConsoleAccessEnabled
--------- --------------------------
account   True
```

The `ManagedBy` field indicates the entity that configured the setting. The possible values are `account` (configured directly) or `declarative-policy`. For more information, see Declarative policies in the *AWS Organizations User Guide*.

Grant account access to the serial console

Console

To grant account access to the serial console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.
4. Choose **Manage**.
5. To allow access to the EC2 serial console of all instances in the account, select the **Allow** checkbox.
6. Choose **Update**.

AWS CLI

To grant account access to the serial console

Use the enable-serial-console-access command.

```
aws ec2 enable-serial-console-access
```

The following is example output.

```
{
    "SerialConsoleAccessEnabled": true
}
```

PowerShell

To grant account access to the serial console

Use the Enable-EC2SerialConsoleAccess cmdlet.

```
Enable-EC2SerialConsoleAccess
```

The following is example output.

```
True
```

Deny account access to the serial console

Console

To deny account access to the serial console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. On the left navigation pane, choose **EC2 Dashboard**.
3. From **Account attributes**, choose **EC2 Serial Console**.
4. Choose **Manage**.
5. To prevent access to the EC2 serial console of all instances in the account, clear the **Allow** checkbox.
6. Choose **Update**.

AWS CLI

To deny account access to the serial console

Use the disable-serial-console-access command.

```
aws ec2 disable-serial-console-access
```

The following is example output.

```
{
    "SerialConsoleAccessEnabled": false
}
```

PowerShell

To deny account access to the serial console

Use the Disable-EC2SerialConsoleAccess cmdlet.

```
Disable-EC2SerialConsoleAccess
```

The following is example output.

```
False
```

Configure IAM policies for EC2 Serial Console access

By default, your users do not have access to the serial console. Your organization must configure IAM policies to grant your users the required access. For more information, see Creating IAM policies in the *IAM User Guide*.

For serial console access, create a JSON policy document that includes the `ec2-instance-connect:SendSerialConsoleSSHPublicKey` action. This action grants a user permission to push the public key to the serial console service, which starts a serial console session. We recommend restricting access to specific EC2 instances. Otherwise, all users with this permission can connect to the serial console of all EC2 instances.

Example IAM policies

- Explicitly allow access to the serial console
- Explicitly deny access to the serial console

- Use resource tags to control access to the serial console

Explicitly allow access to the serial console

By default, no one has access to the serial console. To grant access to the serial console, you need to configure a policy to explicitly allow access. We recommend configuring a policy that restricts access to specific instances.

The following policy allows access to the serial console of a specific instance, identified by its instance ID.

Note that the `DescribeInstances`, `DescribeInstanceTypes`, and `GetSerialConsoleAccessStatus` actions do not support resource-level permissions, and therefore all resources, indicated by the * (asterisk), must be specified for these actions.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
             "Resource": "*"
        },
        {
            "Sid": "AllowinstanceBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

Explicitly deny access to the serial console

The following IAM policy allows access to the serial console of all instances, denoted by the * (asterisk), and explicitly denies access to the serial console of a specific instance, identified by its ID.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DenySerialConsoleAccess",
            "Effect": "Deny",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-
east-1:111122223333:instance/i-0598c7d356eba48d7"
        }
    ]
}
```

Use resource tags to control access to the serial console

You can use resource tags to control access to the serial console of an instance.

Attribute-based access control is an authorization strategy that defines permissions based on tags that can be attached to users and AWS resources. For example, the following policy allows a user to initiate a serial console connection for an instance only if that instance's resource tag and the principal's tag have the same `SerialConsole` value for the tag key.

For more information about using tags to control access to your AWS resources, see Controlling access to AWS resources in the *IAM User Guide*.

Note that the `DescribeInstances`, `DescribeInstanceTypes`, and `GetSerialConsoleAccessStatus` actions do not support resource-level permissions, and therefore all resources, indicated by the `*` (asterisk), must be specified for these actions.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDescribeInstances",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceTypes",
                "ec2:GetSerialConsoleAccessStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowTagBasedSerialConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "ec2-instance-connect:SendSerialConsoleSSHPublicKey"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/SerialConsole":
 "${aws:PrincipalTag/SerialConsole}"
                }
            }
        }
    ]
}
```

Set an OS user password on a Linux instance

You can connect to the serial console without a password. However, to *use* the serial console for troubleshooting a Linux instance, the instance must have a password-based OS user.

You can set the password for any OS user, including the root user. Note that the root user can modify all files, while each OS user might have limited permissions.

You must set a user password for every instance for which you will use the serial console. This is a one-time requirement for each instance.

> **Note**
>
> By default, AMIs provided by AWS are not configured with a password-based user. If you launched your instance using an AMI that already has the root user password configured, you can skip these instructions.

To set an OS user password on a Linux instance

1. Connect to your instance. You can use any method for connecting to your instance, except the EC2 Serial Console connection method.
2. To set the password for a user, use the **passwd** command. In the following example, the user is `root`.

   ```
   [ec2-user ~]$ sudo passwd root
   ```

   The following is example output.

   ```
   Changing password for user root.
   New password:
   ```
3. At the `New password` prompt, enter the new password.
4. At the prompt, re-enter the password.

Connect to the EC2 Serial Console

You can connect to the serial console of your EC2 instance by using the Amazon EC2 console or through SSH. After connecting to the serial console, you can use it for troubleshooting boot,

network configuration, and other issues. For more information about troubleshooting, see Troubleshoot your Amazon EC2 instance using the EC2 Serial Console.

Considerations

- Only 1 active serial console connection is supported per instance.
- The serial console connection typically lasts for 1 hour unless you disconnect from it. However, during system maintenance, Amazon EC2 will disconnect the serial console session.

  The duration of the connection is not determined by the duration of your IAM credentials. If your IAM credentials expire, the connection continues to persist until the maximum duration of the serial console connection is reached. When using the EC2 Serial Console console experience, if your IAM credentials expire, terminate the connection by closing the browser page.
- It takes 30 seconds to tear down a session after you've disconnected from the serial console in order to allow a new session.
- Supported serial console ports: `ttyS0` (Linux instances) and `COM1` (Windows instances)
- When you connect to the serial console, you might observe a slight drop in your instance's throughput.

Topics

- Connect using the browser-based client
- Connect using your own key and SSH client
- EC2 Serial Console endpoints and fingerprints

Connect using the browser-based client

You can connect to your EC2 instance's serial console by using the browser-based client. You do this by selecting the instance in the Amazon EC2 console and choosing to connect to the serial console. The browser-based client handles the permissions and provides a successful connection.

EC2 serial console works from most browsers, and supports keyboard and mouse input.

Before connecting, make sure you have completed the prerequisites.

To connect to your instance's serial port using the browser-based client (Amazon EC2 console)

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2. In the navigation pane, choose **Instances**.
3. Select the instance and choose **Actions**, **Monitor and troubleshoot**, **EC2 Serial Console**, **Connect**.

   Alternatively, select the instance and choose **Connect**, **EC2 Serial Console**, **Connect**.

   An in-browser terminal window opens.
4. Press **Enter**. If a login prompt returns, you are connected to the serial console.

   If the screen remains black, you can use the following information to help resolve issues with connecting to the serial console:

   - **Check that you have configured access to the serial console.** For more information, see Configure access to the EC2 Serial Console.
   - (Linux instances only) **Use SysRq to connect to the serial console.** SysRq does not require that you connect using the browser-based client. For more information, see (Linux instances) Use SysRq to troubleshoot your instance.
   - (Linux instances only) **Restart getty.** If you have SSH access to your instance, then connect to your instance using SSH, and restart getty using the following command.

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```
   - **Reboot your instance.** You can reboot your instance by using SysRq (Linux instances), the EC2 console, or the AWS CLI. For more information, see (Linux instances) Use SysRq to troubleshoot your instance (Linux instances) or Reboot your Amazon EC2 instance.
5. (Linux instances only) At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
6. (Linux instances only) At the `Password` prompt, enter the password, and then press **Enter**.

Connect using your own key and SSH client

You can use your own SSH key and connect to your instance from the SSH client of your choice while using the serial console API. This enables you to benefit from the serial console capability to push a public key to the instance.

After pushing the SSH key to the instance, the SSH connection is not subject to the IAM policies that you configured to grant users EC2 Serial Console access.

Before you begin

Verify that the prerequisites are met.

To connect to an instance's serial console using SSH

1. **Push your SSH public key to the instance to start a serial console session**

   Use the send-serial-console-ssh-public-key command to push your SSH public key to the instance. This starts a serial console session.

   If a serial console session has already been started for this instance, the command fails because you can only have one session open at a time. It takes 30 seconds to tear down a session after you've disconnected from the serial console in order to allow a new session.

   ```
   aws ec2-instance-connect send-serial-console-ssh-public-key \
       --instance-id i-001234a4bf70dec41EXAMPLE \
       --serial-port 0 \
       --ssh-public-key file://my_key.pub \
       --region us-east-1
   ```

2. **Connect to the serial console using your private key**

   Use the **ssh** command to connect to the serial console before the public key is removed from the serial console service. You have 60 seconds before it is removed.

   Use the private key that corresponds to the public key.

   The username format is `instance-id.port0`, which comprises the instance ID and port 0. In the following example, the username is `i-001234a4bf70dec41EXAMPLE.port0`.

   The endpoint of the serial console service is different for each Region. See the EC2 Serial Console endpoints and fingerprints table for each Region's endpoint. In the following example, the serial console service is in the us-east-1 Region.

   ```
   ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-
   connect.us-east-1.aws
   ```

   The following example uses `timeout 3600` to set your SSH session to terminate after 1 hour. Processes started during the session may continue running on your instance after the session terminates.

```
timeout 3600 ssh -i my_key i-001234a4bf70dec41EXAMPLE.port0@serial-console.ec2-instance-connect.us-east-1.aws
```

3. **(Optional) Verify the fingerprint**

   When you connect for the first time to the serial console, you are prompted to verify the fingerprint. You can compare the serial console fingerprint with the fingerprint that's displayed for verification. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, you can confidently connect to the serial console.

   The following fingerprint is for the serial console service in the us-east-1 Region. For the fingerprints for each Region, see EC2 Serial Console endpoints and fingerprints.

   ```
   SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
   ```

   The fingerprint only appears the first time you connect to the serial console.

4. Press **Enter**. If a prompt returns, you are connected to the serial console.

   If the screen remains black, you can use the following information to help resolve issues with connecting to the serial console:

   - **Check that you have configured access to the serial console.** For more information, see Configure access to the EC2 Serial Console.
   - (Linux instances only) **Use SysRq to connect to the serial console.** SysRq does not require that you connect using SSH. For more information, see (Linux instances) Use SysRq to troubleshoot your instance.
   - (Linux instances only) **Restart getty.** If you have SSH access to your instance, then connect to your instance using SSH, and restart getty using the following command.

     ```
     [ec2-user ~]$ sudo systemctl restart serial-getty@ttyS0
     ```

   - **Reboot your instance.** You can reboot your instance by using SysRq (Linux instances only), the EC2 console, or the AWS CLI. For more information, see (Linux instances) Use SysRq to troubleshoot your instance (Linux instances only) or Reboot your Amazon EC2 instance.

5. (Linux instances only) At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
6. (Linux instances only) At the `Password` prompt, enter the password, and then press **Enter**.

EC2 Serial Console endpoints and fingerprints

The following are the service endpoints and fingerprints for EC2 Serial Console. To connect programmatically to an instance's serial console, you use an EC2 Serial Console endpoint. The EC2 Serial Console endpoints and fingerprints are unique for each AWS Region.

Region Name	Region	Endpoint	Fingerprint
US East (Ohio)	us-east-2	serial-console.ec2-instance-connect.us-east-2.aws	SHA256:EhwPkTzRtTY7TRSzz26XbB0/HvV9jRM7mCZN0xw/d/0
US East (N. Virginia)	us-east-1	serial-console.ec2-instance-connect.us-east-1.aws	SHA256:dXwn5ma/xadVMeBZGEru5l2gx+yI5LDiJaLUcz0FMmw
US West (N. California)	us-west-1	serial-console.ec2-instance-connect.us-west-1.aws	SHA256:OHldlcMET8u7QLSX3jmRTRAPFHVtqbyoLZBMUCqiH3Y
US West (Oregon)	us-west-2	serial-console.ec2-instance-connect.us-west-2.aws	SHA256:EMCIe23TqKaBI6yGHainqZcMwqNkDhhAVHa1O2JxVUc
Africa (Cape Town)	af-south-1	ec2-serial-console.af-south-1.api.aws	SHA256:RMWWZ2fVePeJUqzjO5jL2KIgXsczoHlz21Ed00biiWI
Asia Pacific (Hong Kong)	ap-east-1	ec2-serial-console.ap-east-1.api.aws	SHA256:T0Q1lpiXxChoZHplnAkj

Region Name	Region	Endpoint	Fingerprint
			bP7tkm2xXViC9bJFsjYnifk
Asia Pacific (Hyderabad)	ap-south-2	ec2-serial-console.ap-south-2.api.aws	SHA256:WJgPBSwV4/shN+OPITValoewAuYj15DVW845JEhDKRs
Asia Pacific (Jakarta)	ap-southeast-3	ec2-serial-console.ap-southeast-3.api.aws	SHA256:5ZwgrCh+lfns32XITqL/4O0zIfbx4bZgsYFqy3o8mIk
Asia Pacific (Malaysia)	ap-southeast-5	ec2-serial-console.ap-southeast-5.api.aws	SHA256:cQXTHQMRcqRdIjmAGoAMBSExeoRobYyRwT67yTjnEiA
Asia Pacific (Melbourne)	ap-southeast-4	ec2-serial-console.ap-southeast-4.api.aws	SHA256:Avaq27hFgLvjn5gTSShZ0oV7h90p0GG46wfOeT6ZJvM
Asia Pacific (Mumbai)	ap-south-1	serial-console.ec2-instance-connect.ap-south-1.aws	SHA256:oBLXcYmklqHHEbliARxEgH8IsO51rezTPiSM35BsU40
Asia Pacific (Osaka)	ap-northeast-3	ec2-serial-console.ap-northeast-3.api.aws	SHA256:Am0/jiBKBnBuFnHr9aXsgEV3G8Tu/vVHFXE/3UcyjsQ
Asia Pacific (Seoul)	ap-northeast-2	serial-console.ec2-instance-connect.ap-northeast-2.aws	SHA256:FoqWXNX+DZ++GuNTztg9PK49WYMqBX+FrcZM2dSrqrI

Region Name	Region	Endpoint	Fingerprint
Asia Pacific (Singapore)	ap-southeast-1	serial-console.ec2-instance-connect.ap-southeast-1.aws	SHA256:PLFNn7WnCQDHx3qmwLu1Gy/O8TUX7LQgZuaC6L45CoY
Asia Pacific (Sydney)	ap-southeast-2	serial-console.ec2-instance-connect.ap-southeast-2.aws	SHA256:yFvMwUK9lEUQjQTRoXXzuN+cW9/VSe9W984Cf5Tgzo4
Asia Pacific (Thailand)	ap-southeast-7	ec2-serial-console.ap-southeast-7.api.aws	SHA256:KCAZiRYrR1Q2lqsg7vTwixWmvc2wmjVT31XRgSdEfDY
Asia Pacific (Tokyo)	ap-northeast-1	serial-console.ec2-instance-connect.ap-northeast-1.aws	SHA256:RQfsDCZTOfQawewTRDV1t9Em/HMrFQe+CRlIOT5um4k
Canada (Central)	ca-central-1	serial-console.ec2-instance-connect.ca-central-1.aws	SHA256:P2O2jOZwmpMwkpO6YW738FIOTHdUTyEv2gczYMMO7s4
Canada West (Calgary)	ca-west-1	ec2-serial-console.ca-west-1.api.aws	SHA256:s3rc8lI2xhbhr3iedjJNxGAFLPGOLjx7IxxXrGckk6Q

Region Name	Region	Endpoint	Fingerprint
China (Beijing)	cn-north-1	ec2-serial-console.cn-north-1.api.amazonwebservices.com.cn	SHA256:2gHVFy4H7uU3+WaFUxD28v/ggMeqjvSlgngpgLgGT+Y
China (Ningxia)	cn-northwest-1	ec2-serial-console.cn-northwest-1.api.amazonwebservices.com.cn	SHA256:TdgrNZkiQOdVfYEBUhO4SzUA09VWI5rYOZGTogpwmiM
Europe (Frankfurt)	eu-central-1	serial-console.ec2-instance-connect.eu-central-1.aws	SHA256:aCMFS/yIcOdOlkXvOl8AmZ1Toe+bBnrJJ3Fy0k0De2c
Europe (Ireland)	eu-west-1	serial-console.ec2-instance-connect.eu-west-1.aws	SHA256:h2AaGAWO4Hathhtm6ezs3Bj7udgUxi2qTrHjZAwCW6E
Europe (London)	eu-west-2	serial-console.ec2-instance-connect.eu-west-2.aws	SHA256:a69rd5CE/AEG4Amm53I6lkD1ZPvS/BCV3tTPW2RnJg8
Europe (Milan)	eu-south-1	ec2-serial-console.eu-south-1.api.aws	SHA256:lC0kOVJnpgFyBVrxn0A7n99ecLbXSX95cuuS7X7QK30

Region Name	Region	Endpoint	Fingerprint
Europe (Paris)	eu-west-3	serial-console.ec2-instance-connect.eu-west-3.aws	SHA256:q8ldnAf9pymeNe8BnFVngY3RPAr/kxswJUzfrlxeEWs
Europe (Spain)	eu-south-2	ec2-serial-console.eu-south-2.api.aws	SHA256:GoCW2DFRlu669QNxqFxEcsR6fZUz/4F4n7T45ZcwoEc
Europe (Stockholm)	eu-north-1	serial-console.ec2-instance-connect.eu-north-1.aws	SHA256:tkGFFUVUDvocDiGSS3Cu8Gdl6w2uI32EPNpKFKLwX84
Europe (Zurich)	eu-central-2	ec2-serial-console.eu-central-2.api.aws	SHA256:8Ppx2mBMf6WdCw0NUlzKfwM4/IfRz4OaXFutQXWp6mk
Israel (Tel Aviv)	il-central-1	ec2-serial-console.il-central-1.api.aws	SHA256:JR6q8v6kNNPi8+QSFQ4dj5dimNmZPTgwgsM1SNvtYyU
Mexico (Central)	mx-central-1	ec2-serial-console.mx-central-1.api.aws	SHA256:BCuVl13iQNk+CcVnt18Ef4p2ZHUrBBAOxlFetB32GS0

Region Name	Region	Endpoint	Fingerprint
Middle East (Bahrain)	me-south-1	ec2-serial-console.me-south-1.api.aws	SHA256:nPjLLKHu2QnLdUq2kVArsoK5xvPJOMRJKCBzCDqC3k8
Middle East (UAE)	me-central-1	ec2-serial-console.me-central-1.api.aws	SHA256:zpb5duKiBZ+l0dFwPeyykB4MPBYhI/XzXNeFSDKBvLE
South America (São Paulo)	sa-east-1	serial-console.ec2-instance-connect.sa-east-1.aws	SHA256:rd2+/32Ognjew1yVIemENaQzC+Botbih62OqAPDq1dI
AWS GovCloud (US-East)	us-gov-east-1	serial-console.ec2-instance-connect.us-gov-east-1.amazonaws.com	SHA256:tIwe19GWsoyLClrtvu38YEEh+DHIkqnDcZnmtebvF28
AWS GovCloud (US-West)	us-gov-west-1	serial-console.ec2-instance-connect.us-gov-west-1.amazonaws.com	SHA256:kfOFRWLaOZfB+utbd3bRf8OlPf8nGO2YZLqXZiIw5DQ

Disconnect from the EC2 Serial Console

If you no longer need to be connected to your instance's EC2 Serial Console, you can disconnect from it. When you disconnect from the serial console, any shell session running on the instance will continue to run. If you want to end the shell session, you'll need to end it before disconnecting from the serial console.

Considerations

- The serial console connection typically lasts for 1 hour unless you disconnect from it. However, during system maintenance, Amazon EC2 will disconnect the serial console session.
- It takes 30 seconds to tear down a session after you've disconnected from the serial console in order to allow a new session.

The way to disconnect from the serial console depends on the client.

Browser-based client

To disconnect from the serial console, close the serial console in-browser terminal window.

Standard OpenSSH client

To disconnect from the serial console, use the following command to close the SSH connection. This command must be run immediately following a new line.

```
~.
```

The command that you use for closing an SSH connection might be different depending on the SSH client that you're using.

Troubleshoot your Amazon EC2 instance using the EC2 Serial Console

By using EC2 Serial Console, you can troubleshoot boot, network configuration, and other issues by connecting to your instance's serial port.

Use the instructions for your instance's operating system and for the tool you've configured on your instance.

Tools

- (Linux instances) Use GRUB to troubleshoot your instance
- (Linux instances) Use SysRq to troubleshoot your instance
- (Windows instances) Use SAC to troubleshoot your instance

Prerequisites

Before you begin, make sure you have completed the prerequisites, including configuring your chosen troubleshooting tool.

(Linux instances) Use GRUB to troubleshoot your instance

GNU GRUB (short for GNU GRand Unified Bootloader, commonly referred to as GRUB) is the default boot loader for most Linux operating systems. From the GRUB menu, you can select which kernel to boot into, or modify menu entries to change how the kernel will boot. This can be useful when troubleshooting a failing instance.

The GRUB menu is displayed during the boot process. The menu is not accessible via normal SSH, but you can access it using the EC2 Serial Console.

You can boot into single user mode or emergency mode. Single user mode will boot the kernel at a lower runlevel. For example, it might mount the filesystem but not activate the network, giving you the opportunity to perform the maintenance necessary to fix the instance. Emergency mode is similar to single user mode except that the kernel runs at the lowest runlevel possible.

To boot into single user mode

1. Connect to the instance's serial console.
2. Reboot the instance using the following command.

   ```
   [ec2-user ~]$ sudo reboot
   ```

3. During reboot, when the GRUB menu appears, press any key to stop the boot process.
4. In the GRUB menu, use the arrow keys to select the kernel to boot into, and press e on your keyboard.
5. Use the arrow keys to locate your cursor on the line containing the kernel. The line begins with either `linux` or `linux16` depending on the AMI that was used to launch the instance. For Ubuntu, two lines begin with `linux`, which must both be modified in the next step.
6. At the end of the line, add the word `single`.

   The following is an example for Amazon Linux 2.

   ```
   linux /boot/vmlinuz-4.14.193-149.317.amzn2.aarch64 root=UUID=d33f9c9a-\
   dadd-4499-938d-ebbf42c3e499 ro  console=tty0 console=ttyS0,115200n8 net.ifname\
   s=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff rd.she\
   ll=0 single
   ```

7. Press **Ctrl+X** to boot into single user mode.
8. At the `login` prompt, enter the username of the password-based user that you set up previously, and then press **Enter**.
9. At the `Password` prompt, enter the password, and then press **Enter**.

To boot into emergency mode

Follow the same steps as single user mode, but at step 6, add the word `emergency` instead of `single`.

(Linux instances) Use SysRq to troubleshoot your instance

The System Request (SysRq) key, which is sometimes referred to as "magic SysRq", can be used to directly send the kernel a command, outside of a shell, and the kernel will respond, regardless of what the kernel is doing. For example, if the instance has stopped responding, you can use the SysRq key to tell the kernel to crash or reboot. For more information, see Magic SysRq key in Wikipedia.

You can use SysRq commands in the EC2 Serial Console browser-based client or in an SSH client. The command to send a break request is different for each client.

To use SysRq, choose one of the following procedures based on the client that you are using.

Browser-based client

To use SysRq in the serial console browser-based client

1. Connect to the instance's serial console.
2. To send a break request, press `CTRL+0` (zero). If your keyboard supports it, you can also send a break request using the Pause or Break key.

   ```
   [ec2-user ~]$ CTRL+0
   ```

3. To issue a SysRq command, press the key on your keyboard that corresponds to the required command. For example, to display a list of SysRq commands, press `h`.

   ```
   [ec2-user ~]$ h
   ```

The h command outputs something similar to the following.

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

SSH client

To use SysRq in an SSH client

1. Connect to the instance's serial console.
2. To send a break request, press ~B (tilde, followed by uppercase B).

```
[ec2-user ~]$ ~B
```

3. To issue a SysRq command, press the key on your keyboard that corresponds to the required command. For example, to display a list of SysRq commands, press h.

```
[ec2-user ~]$ h
```

The h command outputs something similar to the following.

```
[ 1169.389495] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-
tasks(e) memory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems
(j) sak(k) show-backtrace-all-active-cpus(l) show-memory-usage(m) nice-all-RT-
tasks(n) poweroff(o) show-registers(p) show-all-timers(q) unraw(r
) sync(s) show-task-states(t) unmount(u) show-blocked-tasks(w) dump-ftrace-
buffer(z)
```

Note

The command that you use for sending a break request might be different depending on the SSH client that you're using.

(Windows instances) Use SAC to troubleshoot your instance

The Special Admin Console (SAC) capability of Windows provides a way to troubleshoot a Windows instance. By connecting to the instance's serial console and using SAC, you can interrupt the boot process and boot Windows in safe mode.

> **Note**
>
> If you enable SAC on an instance, the EC2 services that rely on password retrieval will not work from the Amazon EC2 console. Windows on Amazon EC2 launch agents (EC2Config, EC2Launch v1, and EC2Launch v2) rely on the serial console to execute various tasks. These tasks do not perform successfully when you enable SAC on an instance. For more information about Windows on Amazon EC2 launch agents, see the section called "Configure Windows instances". If you enable SAC, you can disable it later. For more information, see Disable SAC and the boot menu.

Tasks

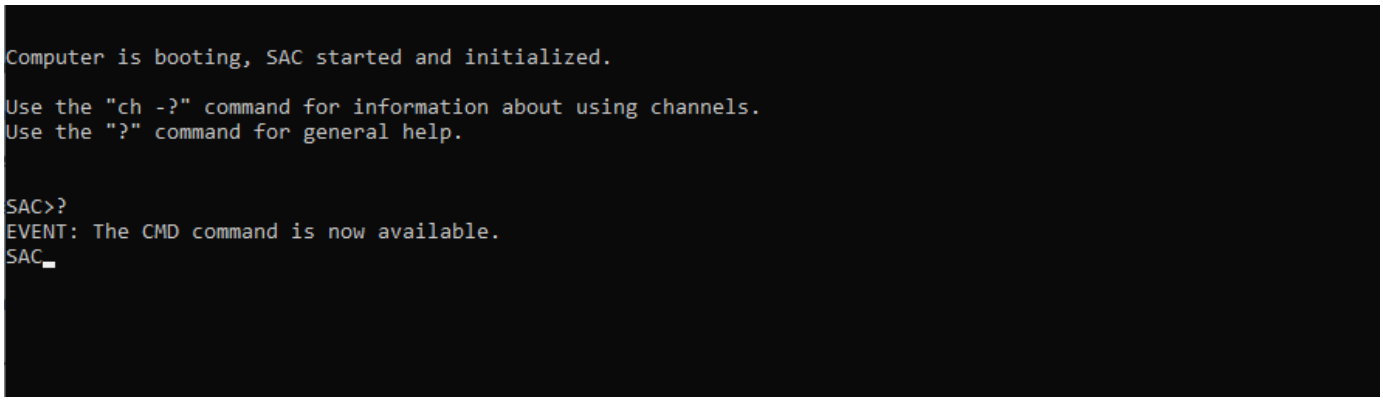
- Use SAC
- Use the boot menu
- Disable SAC and the boot menu

Use SAC

To use SAC

1. Connect to the serial console.

   If SAC is enabled on the instance, the serial console displays the SAC> prompt.



2. To display the SAC commands, enter ?, and then press **Enter**.

   Expected output

```
SAC>?
ch                    Channel management commands.  Use ch -? for more help.
cmd                   Create a Command Prompt channel.
d                     Dump the current kernel log.
f                     Toggle detailed or abbreviated tlist info.
? or help             Display this list.
i                     List all IP network numbers and their IP addresses.
i <#> <ip> <subnet> <gateway> Set IPv4 addr., subnet and gateway.
id                    Display the computer identification information.
k <pid>               Kill the given process.
l <pid>               Lower the priority of a process to the lowest possible.
lock                  Lock access to Command Prompt channels.
m <pid> <MB-allow>    Limit the memory usage of a process to <MB-allow>.
p                     Toggle paging the display.
r <pid>               Raise the priority of a process by one.
s                     Display the current time and date (24 hour clock used).
s mm/dd/yyyy hh:mm    Set the current time and date (24 hour clock used).
t                     Tlist.
restart               Restart the system immediately.
shutdown              Shutdown the system immediately.
crashdump             Crash the system. You must have crash dump enabled.
```

3. To create a command prompt channel (such as `cmd0001` or `cmd0002`), enter cmd, and then press **Enter**.
4. To view the command prompt channel, press **ESC**, and then press **TAB**.

   Expected output

```
Name:                 Cmd0001
Description:          Command
Type:                 VT-UTF8
Channel GUID:         ef9f20a0-1287-11eb-82b0-0e4ba51872e5
Application Type GUID: 63d02271-8aa4-11d5-bccf-00b0d014a2d0

Press <esc><tab> for next channel.
Press <esc><tab>0 to return to the SAC channel.
Use any other key to view this channel.
```

5. To switch channels, press **ESC+TAB+channel number** together. For example, to switch to the `cmd0002` channel (if it has been created), press **ESC+TAB+2**.
6. Enter the credentials required by the command prompt channel.

```
Please enter login credentials.
Username: Administrator
Domain  : .
Password: *******
```

The command prompt is the same full-featured command shell that you get on a desktop, but with the exception that it does not allow the reading of characters that were already output.

```
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>diskpart

Microsoft DiskPart version 10.0.17763.1

Copyright (C) Microsoft Corporation.
On computer: EC2AMAZ-ASR4SAI

DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online           46 GB    46 GB

DISKPART>
```

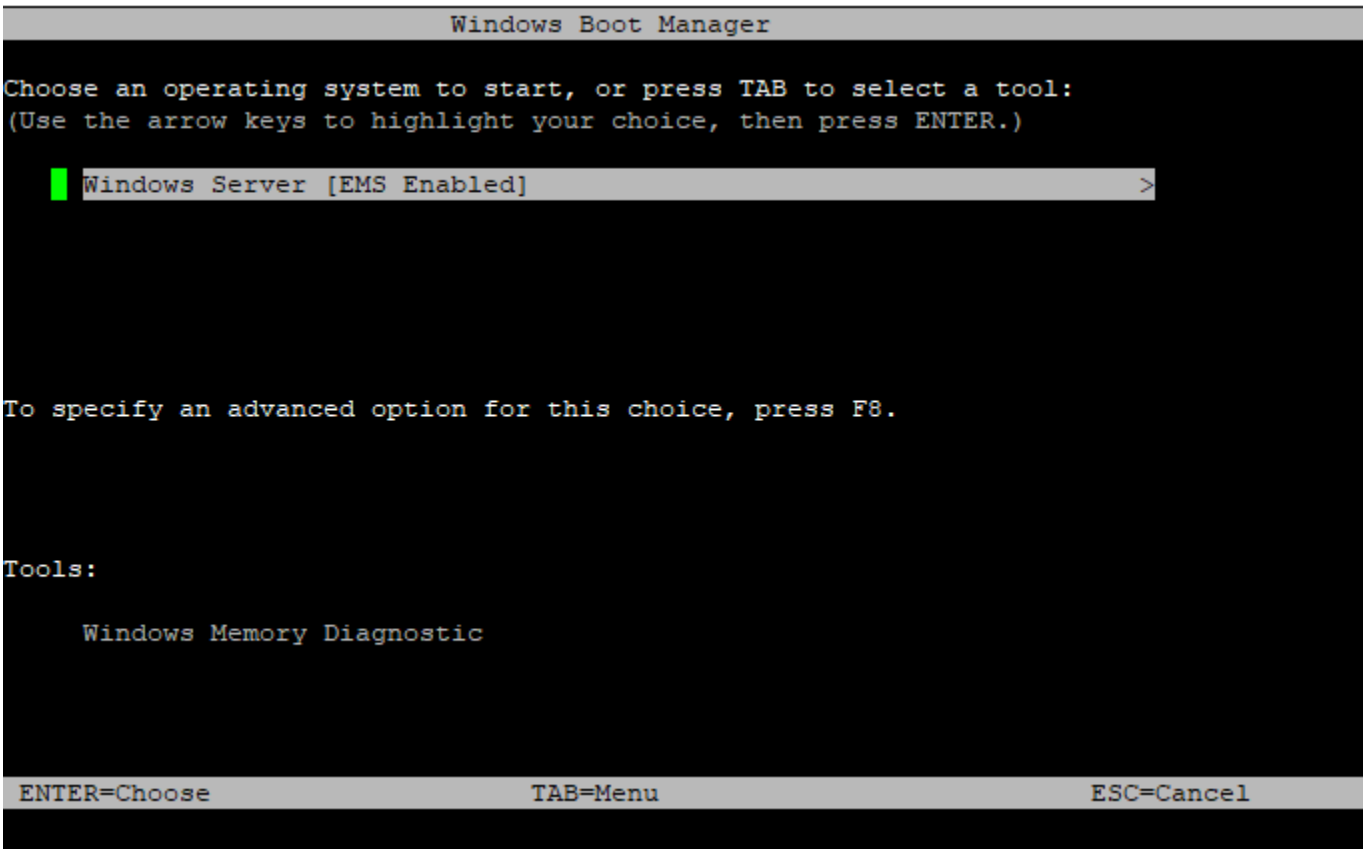
PowerShell can also be used from the command prompt.

Note that you might need to set the progress preference to silent mode.

```
PS C:\Windows\system32> $ProgressPreference="SilentlyContinue"
PS C:\Windows\system32> $computerInfo = Get-ComputerInfo
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Name
Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz
PS C:\Windows\system32> $computerInfo.Csprocessors[0].Description
Intel64 Family 6 Model 85 Stepping 4
PS C:\Windows\system32>
```

Use the boot menu

If the instance has the boot menu enabled and is restarted after connecting through SSH, you should see the boot menu, as follows.



Boot menu commands

ENTER

Starts the selected entry of the operating system.

TAB

Switches to the Tools menu.

ESC

Cancels and restarts the instance.

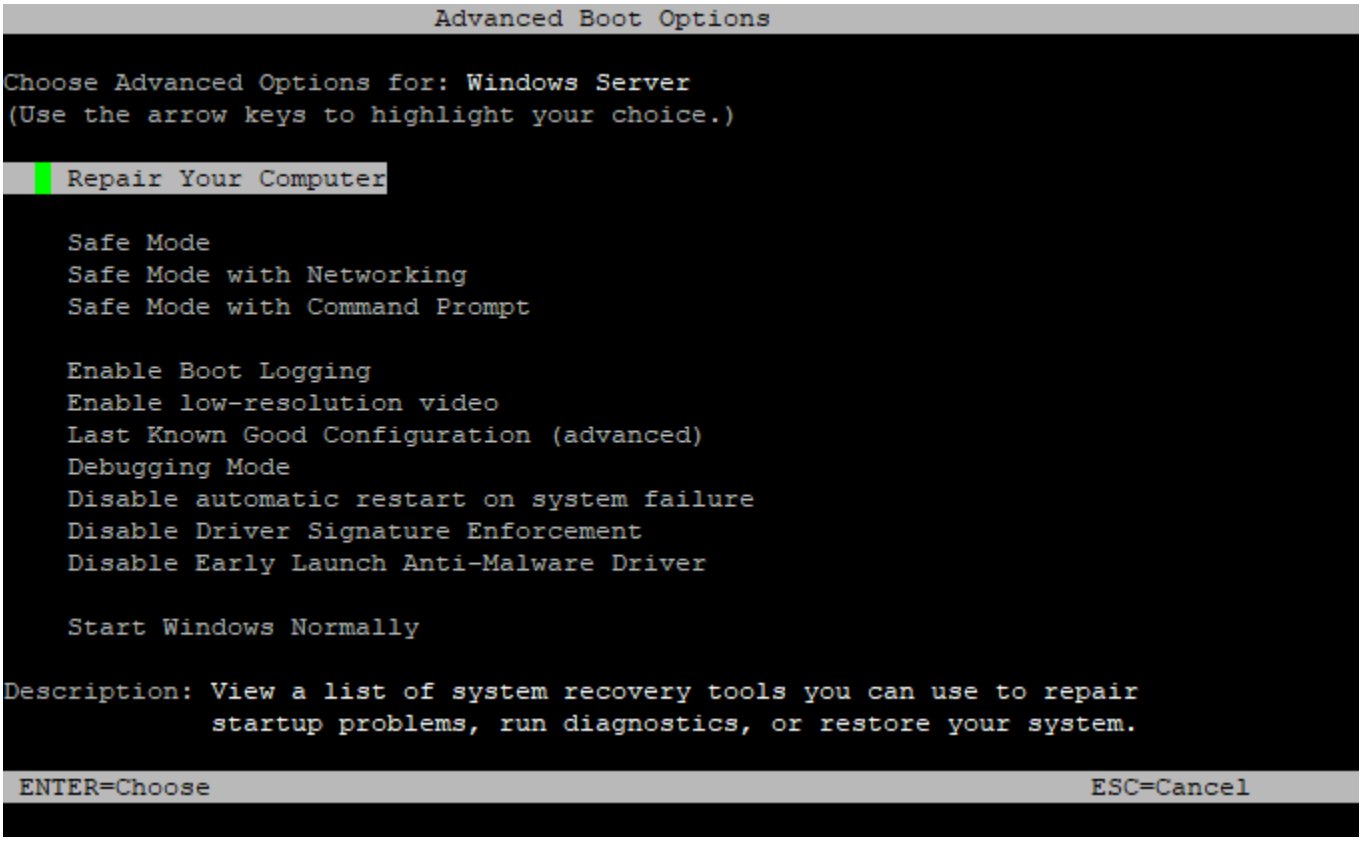
ESC followed by 8

Equivalent to pressing **F8**. Shows advanced options for the selected item.

ESC key + left arrow

Goes back to the initial boot menu.

The ESC key alone does not take you back to the main menu because Windows is waiting to see if an escape sequence is in progress.



Disable SAC and the boot menu

If you enable SAC and the boot menu, you can disable these features later.

Use one of the following methods to disable SAC and the boot menu on an instance.

PowerShell

To disable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from an elevated PowerShell command line.
2. First disable the boot menu by changing the value to no.

   ```
   bcdedit /set '{bootmgr}' displaybootmenu no
   ```

3. Then disable SAC by changing the value to `off`.

   ```
   bcdedit /ems '{current}' off
   ```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Command prompt

To disable SAC and the boot menu on a Windows instance

1. Connect to your instance and perform the following steps from the command prompt.
2. First disable the boot menu by changing the value to no.

```
bcdedit /set {bootmgr} displaybootmenu no
```

3. Then disable SAC by changing the value to `off`.

```
bcdedit /ems {current} off
```

4. Apply the updated configuration by rebooting the instance.

```
shutdown -r -t 0
```

Send a diagnostic interrupt to debug an unreachable Amazon EC2 instance

Warning

Diagnostic interrupts are intended for use by advanced users. Incorrect usage could negatively impact your instance. Sending a diagnostic interrupt to an instance could trigger an instance to crash and reboot, which could lead to the loss of data.

You can send a diagnostic interrupt to an unreachable or unresponsive instance to manually trigger a *kernel panic* for a Linux instance, or a *stop error* (commonly referred to as a *blue screen error*) for a Windows instance.

Linux instances

Linux operating systems typically crash and reboot when a kernel panic occurs. The specific behavior of the operating system depends on its configuration. A kernel panic can also be used to cause the instance's operating system kernel to perform tasks, such as generating a crash dump file. You can then use the information in the crash dump file to conduct root cause analysis and debug the instance. The crash dump data is generated locally by the operating system on the instance itself.

Windows instances

In general, Windows operating systems crash and reboot when a stop error occurs, but the specific behavior depends on its configuration. A stop error can also cause the operating system to write debugging information, such as a kernel memory dump, to a file. You can then use this information to conduct root cause analysis to debug the instance. The memory dump data is generated locally by the operating system on the instance itself.

Before sending a diagnostic interrupt to your instance, we recommend that you consult the documentation for your operating system and then make the necessary configuration changes.

Contents

Supported instance types

Diagnostic interrupt is supported on all Nitro-based instance types, except those powered by AWS Graviton processors. For more information, see instances built on the AWS Nitro System and AWS Graviton.

Prerequisites

Before using a diagnostic interrupt, you must configure your instance's operating system. This ensures that it performs the actions that you need when a kernel panic (Linux instances) or stop error (Windows instances) occurs.

Linux instances

To configure Amazon Linux 2 or Amazon Linux 2023 to generate a crash dump when a kernel panic occurs

1. Connect to your instance.
2. Install **kexec** and **kdump**.

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. Configure the kernel to reserve an appropriate amount of memory for the secondary kernel. The amount of memory to reserve depends on the total available memory of your instance. Open the `/etc/default/grub` file using your preferred text editor, locate the line that starts with `GRUB_CMDLINE_LINUX_DEFAULT`, and then add the `crashkernel` parameter in the following format: `crashkernel=`*`memory_to_reserve`*. For example, to reserve 256MB, modify the `grub` file as follows:

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. Save the changes and close the `grub` file.
5. Rebuild the GRUB2 configuration file.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI. Open the `/etc/sysctl.conf` file using your preferred text editor and add the following.

```
kernel.unknown_nmi_panic=1
```

7. Reboot and reconnect to your instance.
8. Verify that the kernel has been booted with the correct `crashkernel` parameter.

```
$ grep crashkernel /proc/cmdline
```

The following example output indicates successful configuration.

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=256M console=tty0 console=ttyS0,115200n8
 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
 rd.shell=0
```

9. Verify that the **kdump** service is running.

```
[ec2-user ~]$ systemctl status kdump.service
```

The following example output shows the result if the **kdump** service is running.

```
kdump.service - Crash recovery kernel arming
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
 enabled)
   Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
  Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
 Main PID: 2503 (code=exited, status=0/SUCCESS)
```

Note

By default, the crash dump file is saved to `/var/crash/`. To change the location, modify the `/etc/kdump.conf` file using your preferred text editor.

To configure SUSE Linux Enterprise, Ubuntu, or Red Hat Enterprise Linux

On instances based on Intel and AMD processors, the `send-diagnostic-interrupt` command sends an *unknown non-maskable interrupt* (NMI) to the instance. You must configure the kernel to crash when it receives the unknown NMI by adjusting the configuration file for your operating system. For information about how to configure the kernel to crash, see the documentation for your operating system:

- SUSE Linux Enterprise
- Ubuntu
- Red Hat Enterprise Linux (RHEL)

Windows instances

To configure Windows to generate a memory dump when a stop error occurs

1. Connect to your instance.
2. Open the **Control Panel** and choose **System**, **Advanced system settings**.
3. In the **System Properties** dialog box, choose the **Advanced** tab.
4. In the **Startup and Recovery** section, choose **Settings...**.
5. In the **System failure** section, configure the settings as needed, and then choose **OK**.

For more information about configuring Windows stop errors, see Overview of memory dump file options for Windows.

Send a diagnostic interrupt

After you have completed the necessary configuration changes, you can send a diagnostic interrupt to your instance using the AWS CLI or Amazon EC2 API.

AWS CLI

To send a diagnostic interrupt to your instance

Use the send-diagnostic-interrupt command.

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

PowerShell

To send a diagnostic interrupt to your instance

Use the Send-EC2DiagnosticInterrupt cmdlet.

```
Send-EC2DiagnosticInterrupt -InstanceId i-1234567890abcdef0
```