Canvas - Linux Fundamentals -Ticket Challenge - Interns

Case 3.1

Hi Support, my instance is showing that there is no space left on the device. I increased the size of my EBS volume in the AWS console page but on the instance it is still showing the old size. Please help resolve this issue.

Hi Customer Name
Thank you for reaching out. My name is Abreham Getachew and I will be assisting you with some steps to increasing the disk space for your EBS volume. I understand that you've increased the size of your EBS volume, but your instance is still showing no additional space. This issue usually occurs because the filesystem on the instance has not been extended to utilize the new volume size.
To resolve this issue, you need to extend the filesystem on your instance. AWS provides detailed instructions on how to do this in their documentation. Please follow the steps outlined in the following guide:

AWS Documentation: Resizing the EBS Volume and Filesystem

I know you have increased the volume size but incase you need reference you can follow the instructions to increase the volume size in the AWS Management Console: Modify an EBS Volume

Extend the Filesystem on the Instance:

- After modifying the volume, follow these steps to extend the filesystem:
  - For Linux instances, refer to: Extending a Linux File System
  - For Windows instances, refer to: Extending a Windows File System

The Linux guide includes commands for checking the disk space, verifying the new volume size, and resizing the filesystem. Please follow the instructions carefully to ensure that your instance recognizes the expanded storage.
If you encounter any issues or need further assistance, please let me know, and I'll be happy to help.

Best regards,
Abreham Getachew
AWS Support Engineer

Case 3.2

Hello AWS Support, I am not being able to ssh to EC2 instances in any location! Please help! Kind regards, Kimberly

Response:
Hello Kimberly,

Thank you for reaching out to AWS. My name is Koi and I will be assisting you today about your query. I understand you are having difficulties SSH into your Ec2 instances.

To troubleshoot this issue, please provide a screenshot of the error message you are receiving when SSH into your instances. Can you provide more details about how you are trying to connect to the instance, including the commands used?

There are a couple things to check when diagnosing a SSH problem. Here are several reasons why you are unable to SSH into your Ec2 instance:

1. Check that your instance status is in the running state and not stopped. Go to the Ec2 dashboard and look at the column “Instance State”. If it is not in the running state then start your instance before SSH.
2. Confirm that your Ec2 instance’s public IP address has not changed. (it can change when the instance is stopped and started) A way to fix this to keep a stagnant public IP address would be to create an elastic IP address. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html
3. Check that the instance is in a public subnet. In order for the route table to point to the internet gateway. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#TroubleshootingInstancesConnectionTimeout
4. Check the security group settings: Make sure that your security group allows incoming traffic on port 22. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#TroubleshootingInstancesConnectionTimeout
5. Check the network ACL settings: Network ACLs are another layer of security in your VPC. Make sure that your network ACLs allow incoming traffic on port 22.https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#TroubleshootingInstancesConnectionTimeout
6. If you are receiving a “connection timed out” or “connection refused” error message. This error message comes from the SSH client. The error indicates that the server didn't respond to the client and the client program gave up (timed out). https://repost.aws/knowledge-center/ec2-linux-resolve-ssh-connection-errors

Best regards,
Koi Lindstrom
AWS Support Engineer

Case 3.3

Customer Scenario 3
Hello AWS Support, My Linux instance has failed its instance check after upgrade and reboot and I cannot connect to it. i-0436f85533745cd8b showing status check 1/2 hence we are unable to access this instance. I’ve tried rebooting and nothing is helping me out.. Thanks in advance. Joe
Key items to keep in mind: How can you view log messages for EC2 instances?

Response:

Dear Joe,

Thank you for reaching out to AWS Support regarding the issue with your Linux instance failing its instance status checks after an upgrade and reboot. My name is Gururaj, I will be assisting you today. I understand the problem of not being able to access your instance, and I'll do my best to assist you in resolving this problem.
When an instance fails one or more instance status checks, it typically indicates an underlying issue with the instance itself. In such cases, it's essential to review the log messages to identify the root cause of the problem.

To view log messages for EC2 instances, you can follow these steps:

- Connect to the instance using Instance Connect:
- Open the Amazon EC2 console and navigate to your instance.
- Click on the "Connect" button and choose the "Instance Connect" option.
- Follow the prompts to connect to your instance securely without an SSH key pair.

View system log files:

- Once connected to the instance, you can view various system log files to identify any error messages or relevant information.
- For Linux instances, some common log files to check are:

  /var/log/messages: General system messages **tail -n 100 /var/log/messages** (shows the last 100 lines)
  /var/log/cloud-init-output.log: Cloud-init logs **less /var/log/cloud-init-output.log** (allows scrolling through the file)
  /var/log/syslog: System log messages **tail /var/log/syslog** (shows the last lines)
  /var/log/kern.log: Kernel log messages l**ess /var/log/kern.log** (allows scrolling through the file)
  /var/log/dmesg: Kernel ring buffer messages **dmesg** (displays the kernel ring buffer)

- You can view these log files using commands like tail or less. For example, tail -n 100 /var/log/messages will show the last 100 lines of the /var/log/messages file.

References:

- Viewing Output from Instance System Log

  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/troubleshoot-unreachable-instance.html

- Monitoring Instances Using CloudWatch

  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch.html

- Troubleshooting Instance Status Checks

  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstances.html

Regards
Gururaj Hanamesh Desai
AWS Support Engineer

Case 3.4

Good day, I am facing issues ssh'ing into my EC2 linux instance as the private key is unprotected. Please explain what this means and provide guidance on how to resolve the issue? Thanks, Steven

Hello Steven,

Thank you for contacting AWS Support, I hope you're doing well. I understand that you are facing issues ssh'ing into your EC2 linux instance. An EC2 instance needs a private key to be properly secured. This means that the permissions on your private key file are too open and it can be read or written by others, which SSH is considering a security risk. In order to ssh successfully, secure your private key[1].

Firstly, double check you're using the right key for your instance. To resolve this issue, open a terminal on your local machine as you normally would.

For a Mac [2]: Get your private key file (usually ends with ".pem") by navigating to the directory that the key is stored in. Then run "`[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem`". This makes it so that your permissions on the private key file are only accessible (for read and write) for you, the user.

For connecting from Windows[2]:

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 via SSH.

For Windows Command Prompt[2]: Use "`icacls.exe $path /reset"` to reset any explicit permissions. Then run "`icacls.exe $path /GRANT:R "$($env:USERNAME):(R)"` to update permissions. After updating the permissions, you should be able to SSH into your EC2 instance. I have attached documentation below for your reference. If you have any further questions or issues, please let us know.

Reference:
[1] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html#troubleshoo t-unprotected-key
[2] https://repost.aws/knowledge-center/ec2-server-refused-our-key

Sincerely,
KT Lamessa
AWS Support Engineer

Case 3.5

Hello there,

Thank you for contacting AWS Premium Support today. My name is Jason Zhang, and it's a pleasure to work with you. I understand that you're experiencing performance issues with your Linux instances, which are slow and unresponsive. Several factors, such as CPU, memory, and disk, can contribute to these issues. Let's start by checking our CPU utilization.

Steps to Check CPU Utilization:
Option 1: Via EC2 Instance Console:
Go to the EC2 console.
Navigate to your Linux instance and select the “Monitoring” tab (EC2 → Your Linux Instance → Monitoring tab).
Check the “CPU Utilization” metric, which might be the cause of the slowness and unresponsiveness.
If the CPU usage is consistently high, verify the current instance type from the “Detail” tab. If the instance type is as expected but performance issues persist, consider upgrading to a more powerful instance type [1]. For deeper analysis, use the `top` command on your Linux instance to check for any anomalous processes consuming excessive resources [2].

Option 2: Using CloudWatch:
CloudWatch collects and processes data from EC2 instances into near real-time metrics and you can check the utilization metrics for your Linux instance directly from CloudWatch [3]. To prevent CPU utilization from reaching 100% and affecting your business, create a CloudWatch alarm to notify you when the CPU metric hits a certain threshold [4]. This will give you enough time to make necessary adjustments before the instance becomes slow and unresponsive.

By following these steps, you should get a good overview of your Linux instance's performance and identify potential issues causing the slowdown.

Please feel free to reach out if you have any questions. If needed, we can arrange a call to discuss further.

Your feedback is important to us. Please share your experience by rating this response. You will find a link to the AWS Support Center at the end of this correspondence to rate us.

Best regards,
Jason Zhang
AWS Support Engineer

Reference links:
[1] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-resize.html
[2 - third party]https://www.geeksforgeeks.org/top-command-in-linux-with-examples/
[3] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch.html
[4] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch-createalarm.html