Blog / Amazon Web Services

Amazon EBS volumes: How to Shrink ’em Down to Size

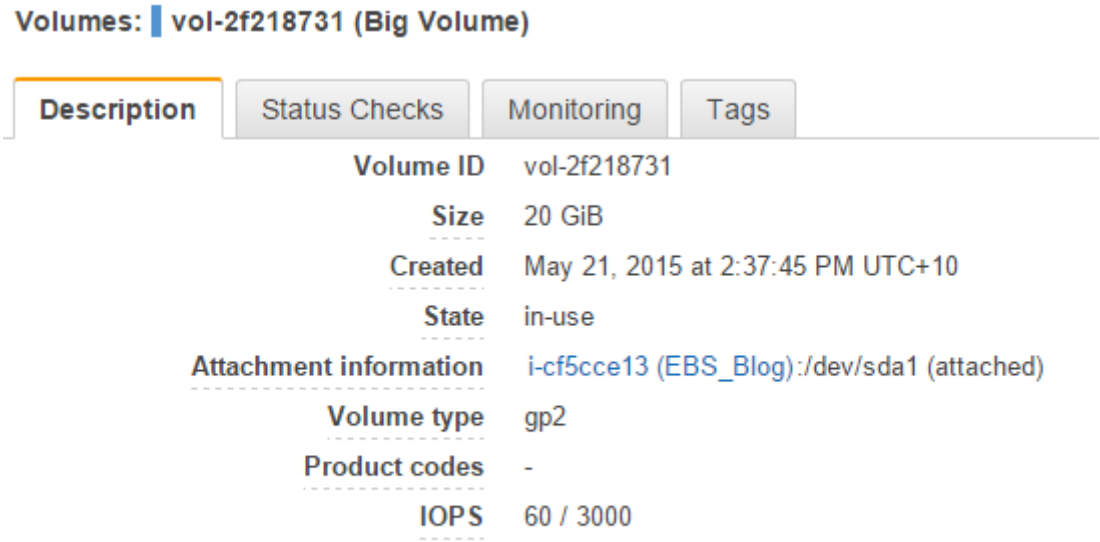
MICHAEL SHEEHY – MAY 26, 2015

Have you ever launched an EC2 instance with Amazon EBS (Elastic Block Store) storage only to realize that your EBS storage is way over-allocated and you have no idea on how to reduce it?

Amazon’s Elastic Block Store Volumes are easy to use and expanding them is no problem at all, but for some reason, there’s no obvious way to reduce them. This is especially problematic when the EBS volume is mounted on the root.

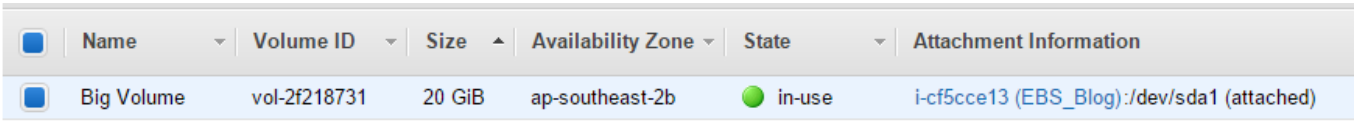
It turns out that there’s nothing to worry about. I’ll take you through an easy way to reduce your volumes so you can hopefully save yourself some money.

Reducing AWS EBS Volumes



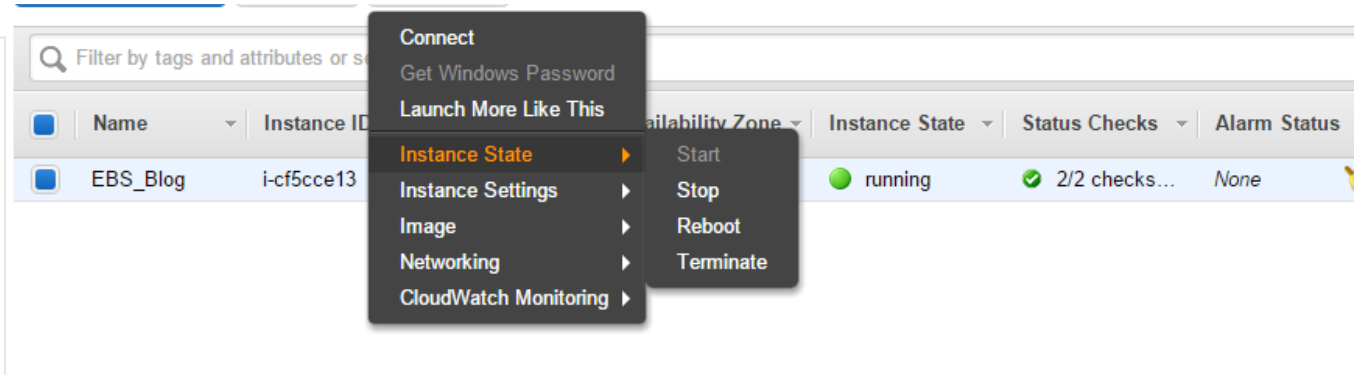
For the purpose of this exercise, I created an instance and launched it with a 20GB Amazon EBS volume:

Assuming we want to reduce this to 8GB, the first thing we will need to do is to make a note of the root volume's block device name and our instance's availability zone

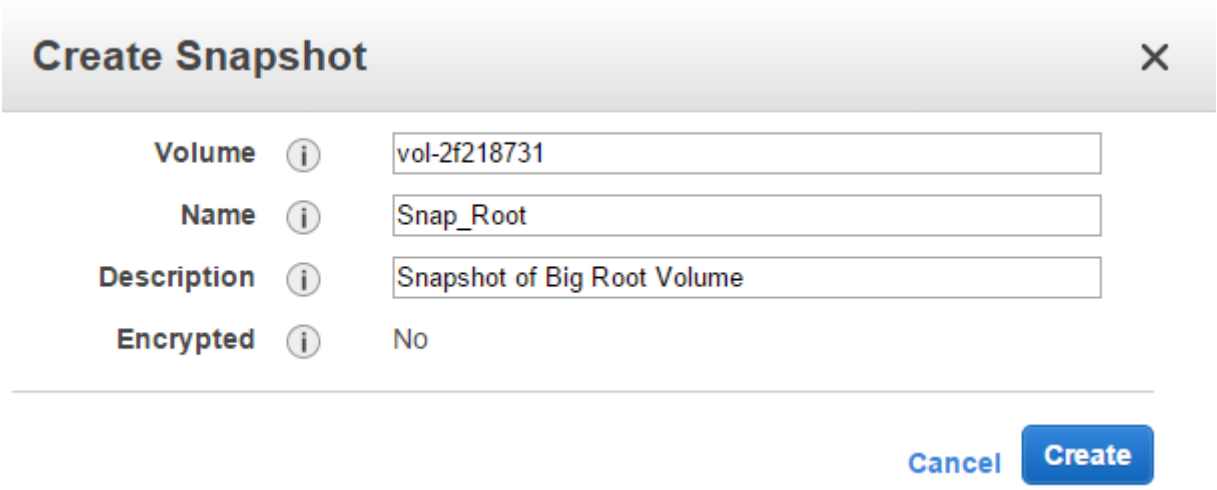


So in my case, the details are
Block Device Name = */dev/sda1*
Availability Zone = *ap-southeast-2b*

Stop the Instance:

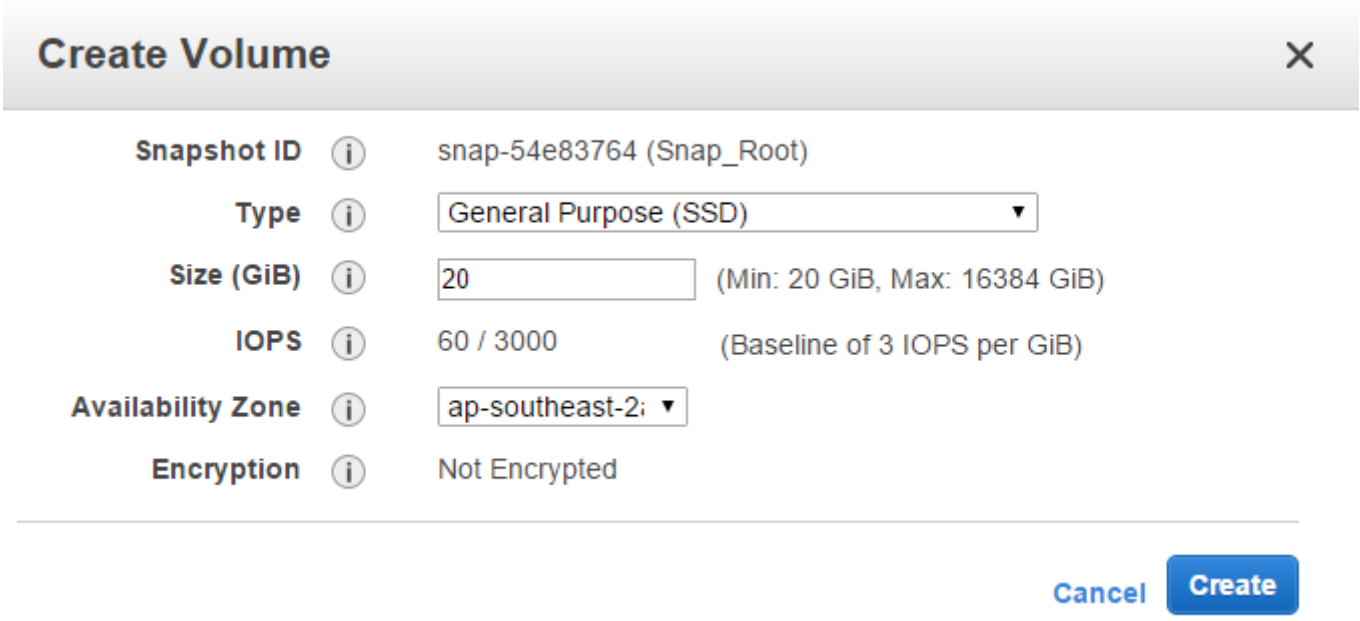


Create a snapshot of the root volume:

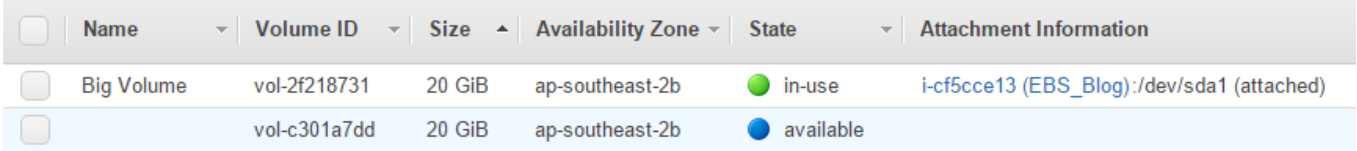


Create a second Amazon EBS volume:

Using the snapshot, create a second volume – of the original size – in the same availability zone as your instance.



Hopefully, you will have something like this:

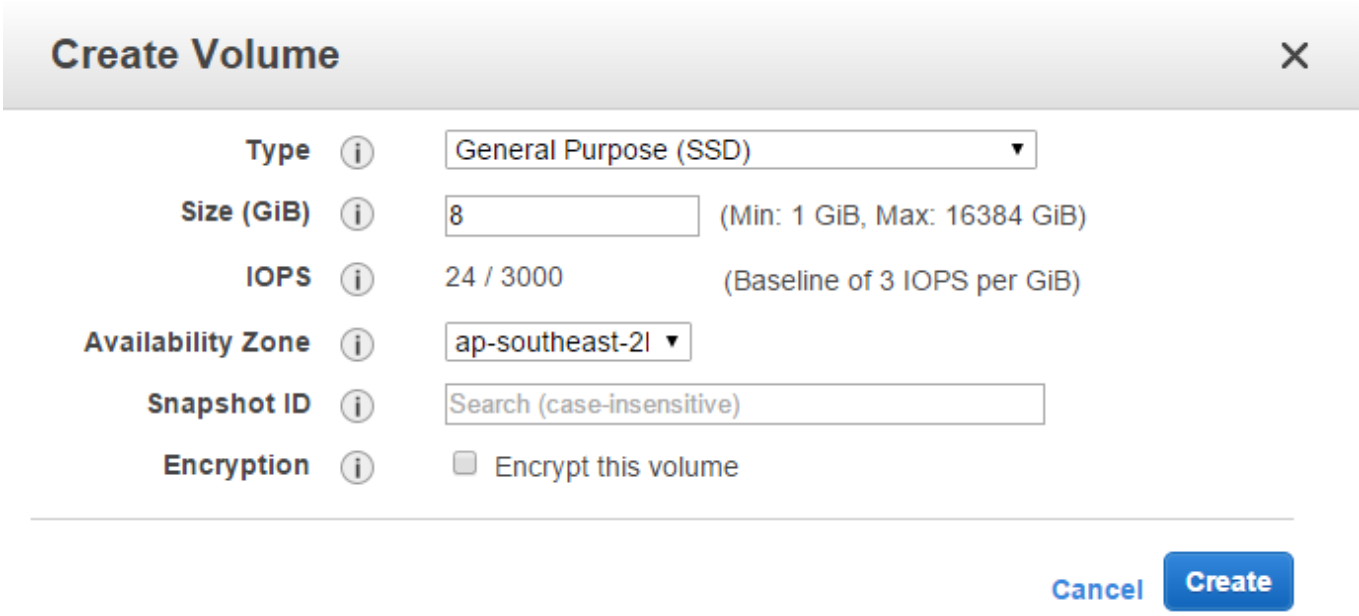


Name	Volume ID	Size	Availability Zone	State	Attachment Information
Big Volume	vol-2f218731	20 GiB	ap-southeast-2b	in-use	i-cf5cce13 (EBS_Blog):/dev/sda1 (attached)
	vol-c301a7dd	20 GiB	ap-southeast-2b	available	

Create an empty 8GB Amazon EBS volume in the same availability zone

Name	Volume ID	Size	Availability Zone	State	Attachment Information
Small Volume	vol-2c0caa32	8 GiB	ap-southeast-2b	in-use	i-cf5cce13 (EBS_Blog):/dev/sdf (attached)
Big Volume	vol-2f218731	20 GiB	ap-southeast-2b	in-use	i-cf5cce13 (EBS_Blog):/dev/sda1 (attached)
Big Volume Snapshot	vol-c301a7dd	20 GiB	ap-southeast-2b	in-use	i-cf5cce13 (EBS_Blog):/dev/sdg (attached)

attach both volumes to the instance and again note all device name details.



Now, attach both volumes to the instance and again note all device name details.

Block Device Name Big Volume = */dev/sda1*
Block Device Name Big Volume Snapshot = */dev/sdg*
Block Device Name Small Volume = */dev/sdf*

Restart the Instance and SSH in

Login:

```
ssh -i <Private-key> ec2-user@ip-address
```

Create a file system for the 2 volumes you have created (Note: In Ubuntu I had to do a *cat/proc/partitions* to work out which device was which).

```
sudo mkfs -t ext4 /dev/xvdf
sudo mkfs -t ext4 /dev/xvdg
```

Create two mount directories and mount the new volumes.

```
sudo mkdir /mnt/small
sudo mount /dev/xvdf /mnt/small
```

```
sudo mkdir /mnt/snap
sudo mount /dev/xvdg1 /mnt/snap
```

Sync the files.

```
sudo rsync -aHAXxSP /mnt/snap/ /mnt/small
```

Unmount the smaller volume.

```
sudo umount /dev/xvdf
```

Stop the instance

Detach all volumes.

Name	Volume ID	Size	Availability Zone	State	Att
Small Volume	vol-2c0caa32	8 GiB	ap-southeast-2b	available	
Big Volume	vol-2f218731	20 GiB	ap-southeast-2b	available	
Big Volume Snapshot	vol-c301a7dd	20 GiB	ap-southeast-2b	available	

Attach the small volume to the Block Device Name from the first step.
Block Device Name = */dev/sda1*

Name	Volume ID	Size	Availability Zone	State	Attachment Information
Small Volume	vol-2c0caa32	8 GiB	ap-southeast-2b	in-use	i-cf5cce13 (EBS_Blog):/dev/sda1 (attached)
Big Volume	vol-2f218731	20 GiB	ap-southeast-2b	available	
Big Volume Snapshot	vol-c301a7dd	20 GiB	ap-southeast-2b	available	

Now you can restart your instance and verify that it is working correctly.

I recommend taking a look at Cloud Academy's Managing Instance Volumes Using EBS Hands-on Lab if you want to learn to create an EC2 instance with an additional EBS volume.

IMPORTANT Don't forget to delete your snapshot and the two other volumes that are no longer needed. That can save you some big bucks.

AWS EBS

WRITTEN BY
Michael Sheehy

I have been UNIX/Linux System Administrator for the past 15 years and am slowly moving those skills into the AWS Cloud arena. I am passionate about AWS and Cloud Technologies and the exciting future that it promises to bring.

cat/proc/partitions to work out which device was which).
sudo mkfs -t ext4 /dev/xvdf
sudo mkfs -t ext4 /dev/xvda1

Can someone explain this !

61 • Reply • Share ›

Theo Bittencourt ➚ Naresh Kumar • 4 years ago

I lost 1 hour because this mistake. Unfun. Fix it please.

3 • Reply • Share ›

Mike Muldoon ➚ Naresh Kumar • 4 years ago

Yeah, I had the same thought. That looks like a typo.

1 • Reply • Share ›

Ernesto Tagwerker ➚ Mike Muldoon • 3 years ago

It's definitely a typo. If you do `mkfs` on the snapshot and then `rsync` it won't copy anything because you wiped all the info from the snapshot.

I followed the instructions and then I saw this comment. :(

• Reply • Share ›

Show more replies

Ernesto Tagwerker • 3 years ago

Hi Michael,

Your guide got me 90% there. Unfortunately it didn't work all the way. The instance wouldn't boot with the new (downsized) volume.

I tried this alternative approach: https://matt.berther.io/201... and it worked for me.

I'm just posting this here in case someone else has problems with your steps.

Thanks for publishing this article!

7 • Reply • Share ›

Rogério Chaves • 4 years ago

Thanks, it was very helpful, but after doing that I stuck at initializing, and when I opened the system logs I saw this error:

“block:/dev/disk/by-label/\x2f”

After some hours I finally understood that I should label my new volume with the same as the old one, using e2label (http://www.cyberciti.biz/fa..., this command worked for me:

sudo e2label /dev/sdf /

2 • Reply • Share ›

hibbelig • 3 years ago

It seems that some distros use the UUID to mount the file system. When that is the case,

after copying from the snapshot to the smaller filesystem, edit /boot/grub/grub.cfg and /etc/fstab on the smaller filesystem to reflect the new UUID.

Actually, I'm unclear about handling /boot/grub/grub.cfg. In Ubuntu, that's auto-generated, but I wasn't able to find the magic incantations that would generate it correctly for me -- of course I needed to do this while the new disk was not running yet.

1 • Reply • Share ›

Sailesh Jaiswal • 2 years ago

I've shrink the volume to 100GB to 20GB. Followed above steps and now I am not able to start the instance.. status: Client.InstanceInitiatedShutdown: Instance initiated shutdown

did I lost all data? how to fix.

• Reply • Share ›

Michael Howarth • 2 years ago

Thank you for the clearly written tutorial, however when I get to the parts where I need to "Create a file system for the 2 volumes you have created", I have a problem:

Making a file system on the clean, new drive works fine, but when trying to make a file system on the snapshot of the root I get the following result.

sudo mkfs -t ext4 /dev/xvdg
mke2fs 1.42.13 (17-May-2015)
Found a dos partition table in /dev/xvdg
Proceed anyway? (y,n) y
/dev/xvdg is apparently in use by the system; will not make a filesystem here!

"xvdg" is the snapshot of the root and isn't actually used to boot with so I'm a little confused here.

If I make an assumption (based on the top comment on this blog) that there's a mistake in the blog and I don't need to create a file system on this disk, then go straight on to the next bit - Create two mount directories and mount the new volumes

see more

• Reply • Share ›

Alper Alimoglu • 2 years ago

I have followed following guide based on the answer in order. I was not able to do following command (sudo mkfs -t ext4 /dev/xvdg).

[$] sudo mkfs -t ext4 /dev/xvdg //I am not able to accomplish this command<========

mke2fs 1.42.13 (17-May-2015)
Found a dos partition table in /dev/xvdg
Proceed anyway? (y,n) y
/dev/xvdg is apparently in use by the system; will not make a filesystem here!

In detail: https://stackoverflow.com/q...

• Reply • Share ›

Alaa Attya • 2 years ago

i've followed all those steps but the instance would not start. it's turing to stop state.

BTW i did not format the volume that was created from the snap
anyone have any clue?

Reply • Share ›

AR • 4 years ago

there is a different approach that might be better at preserving low-level bits of the filesystem, such as timestamps etc - https://matt.berther.io/201...

Reply • Share ›

paul thomas • 4 years ago

Followed this to a T, couldn't start off the instance. Set an e2label no joy.

Reply • Share ›

Michael Sheehy • 4 years ago

Hi Deep,

There is still an error in the post that we will fix soon that may have something to do with your problem.

This line below needs to change

sudo mount /dev/xvda1 /mnt/snap

should be

sudo mount /dev/xvdg /mnt/snap

Reply • Share ›

paul thomas ➚ Michael Sheehy • 4 years ago

your example has

sudo mount /dev/xvdg1 /mnt/snap

1 Reply • Share ›

Clement Kerneur ➚ Michael Sheehy • 2 years ago

Thanks Michael, I confirm this solution it's work

the small device should have the label / to boot correctly.
else the instance is stuck in "Initializing"

• Reply • Share ›

Show more replies

Deep • 4 years ago

Managed to go a bit further but now stuck at this.

• Reply • Share ›

Deep • 4 years ago

My Instance is stuck in "Initializing"

I had 2 volumes attached one is root and one another.

I wanted to chop down the root.

So created a snapshot of root and then a new volume from it, created another smaller volume.

Attached these two.

Created Filesystem and mounted,synced and unmounted.

After that detached the old root and attached the new root.

Any help ?

Last traces in system logs :

---

see more

• Reply • Share ›

Michael Sheehy • 4 years ago

Hi Naresh,

I suspect I made a mistake in the mappings.

When you attach an EBS volume, you specify the device to attach it as. Under linux, these devices are /dev/xvd* - and are symlinked to /dev/sd*

If you look at the contents of dev it should be something similar to below

ls -l /dev/sd* /dev/xv*
lrwxrwxrwx 1 root root 5 Dec 12 18:32 /dev/sda1 -> xvda1
lrwxrwxrwx 1 root root 4 Dec 12 18:32 /dev/sdf -> xvdf
lrwxrwxrwx 1 root root 4 Dec 12 18:32 /dev/sdg -> xvdg
brw-rw---- 1 root disk 202, 1 Dec 12 18:32 /dev/xvda1
brw-rw---- 1 root disk 202, 80 Dec 12 18:32 /dev/xvdf
brw-rw---- 1 root disk 202, 96 Dec 12 18:32 /dev/xvdg

So the file systems we need to create are

sudo mkfs -t ext4 /dev/xvdf
sudo mkfs -t ext4 /dev/xvdg

NOT

sudo mkfs -t ext4 /dev/xvdf
sudo mkfs -t ext4 /dev/xvda1

i will change it in the article to reflect this

 • Reply • Share ›

geeteerebel ➚ Michael Sheehy • 3 years ago

just run lsblk and you'll see everything..

 • Reply • Share ›

Search

RELATED TRAINING CONTENT

- **Managing Instance Volumes Using EBS**
- **Expanding EBS Volumes on Linux Instances**
- **Expanding EBS Volumes on Windows Instances**
- **Using Amazon Key Management Service to Encrypt S3 and EBS Data**
- **Amazon EKS - Launch Kubernetes Cluster and Deploy Wordpress**


Amazon EC2
Container Service
and Elastic
Beanstalk: Docker

Amazon EC2 Container Service and Elastic Beanstalk: Docker on AWS

Technology keeps moving. In just a few years, we've gone from servers running on dedicated hardware, through virtualization, and then cloud computi...

Learn More

AWS: RAID 0 Configuration on EBS Volumes

Amazon Web Services EBS Volumes provides the block level storage volumes of Amazon EC2 instances. EBS volumes are highly available and reliable sto...

Learn More

How to Encrypt an EBS Volume

Keeping data and applications safe in the cloud is one the most visible challenges facing cloud teams in 2018. Cloud storage services where data re...

Learn More

GET REGULAR UPDATES

Email

Subscribe

Related Posts

SAM GHARDASHEM — MAY 15, 2019

Aviatrix Integration of a NextGen Firewall in AWS Transit Gateway

Learn how Aviatrix's intelligent orchestration and control eliminates unwanted tradeoffs encountered when deploying Palo Alto Networks VM-Series Firewalls with AWS Transit Gateway.Deploying any next generation firewall in a public cloud environment is challenging, not because of the f...

Read more

Amazon Web Services AWS

JOE NEMER — MAY 3, 2019

AWS Config Best Pra Compliance

Use AWS Config the Right W ComplianceIt's well-known t a powerful service for monito across your resources. As AW constantly evolved and impro years, it has transformed into powerhouse for monitoring y

Read more

AWS Compliance

SOLUTIONS

Managing the Multi-Cloud

Migrate to the Cloud

Develop Tech Talent

Optimize Cloud Spend

CONTENT LIBRARY

Amazon Web Services

Microsoft Azure

Google Cloud Platform

DevOps

Cloud Migration

Security

Serverless

Big Data

Machine Learning

Cloud Computing Fundamentals

Certifications

Partner Ecosystem

Upcoming Content

RESOURCES

Blog

Cloud Roster™

Cloud Catalog™

Events

Newsroom

Webinars

Case Studies

Whitepapers

Support

AWS Certifications

Azure Certifications

Google Cloud Certifications

Courses

Release Notes

COMPANY

About

Careers We are hiring!

Partners

Contact Us

Instructors