If EC2's EBS volume storage exceeded more than 75%, get an alert /alarm using CloudWatch and Terraform.

1. Installs/starts the **CloudWatch Agent** on your existing EC2 so the instance actually **reports disk usage** (the metric disk\_used\_percent).
2. Uses **Terraform** to create an **SNS topic** + **CloudWatch alarm** that fires when root / usage > 75%.

**Overview of steps**

1. Give the EC2 instance an IAM instance profile with the **CloudWatchAgentServerPolicy** so it can publish metrics.
2. Install & configure the **Amazon CloudWatch Agent** on the EC2 to collect disk\_used\_percent for /.
3. Use **Terraform** to create an **SNS topic** (email subscription) and an **aws\_cloudwatch\_metric\_alarm** on CWAgent/disk\_used\_percent with dimensions InstanceId + path="/".
4. Verify the metric appears in CloudWatch and test the alarm.

**A — Add IAM permissions to the EC2 (two options)**

**Option A — (Console: easiest if you prefer UI)**

1. Go to **IAM → Roles → Create role**.
2. Select **AWS service → EC2**.
3. Attach managed policy **CloudWatchAgentServerPolicy**. Create role (name it e.g. CloudWatchAgentServerRole).
4. Go to **EC2 → Instances → Actions → Security → Modify IAM role** and attach the new role to your instance.
5. Wait a minute for role to propagate.

**Option B — (AWS CLI: reproducible)**

Create a trust file trust.json:

{

"Version":"2012-10-17",

"Statement":[

{

"Effect":"Allow",

"Principal":{"Service":"ec2.amazonaws.com"},

"Action":"sts:AssumeRole"

}

]

}

Commands (replace names as desired):

aws iam create-role --role-name CloudWatchAgentServerRole \

--assume-role-policy-document file://trust.json

aws iam attach-role-policy --role-name CloudWatchAgentServerRole \

--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy

aws iam create-instance-profile --instance-profile-name CloudWatchAgentProfile

aws iam add-role-to-instance-profile --instance-profile-name CloudWatchAgentProfile \

--role-name CloudWatchAgentServerRole

aws ec2 associate-iam-instance-profile --instance-id i-0123456789abcdef0 \

--iam-instance-profile Name=CloudWatchAgentProfile

Wait a minute after the associate-iam-instance-profile call.

**B — Install and configure the CloudWatch Agent on the EC2**

SSH into the EC2 instance and run the commands below depending on your distro.

**Install the agent**

Amazon Linux 2:

sudo yum install -y amazon-cloudwatch-agent

Amazon Linux 2023 / RHEL/CentOS (if dnf):

sudo dnf install -y amazon-cloudwatch-agent

**Create a minimal agent config that monitors /**

Create /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json with this content:

{

"metrics": {

"namespace": "CWAgent",

"metrics\_collected": {

"disk": {

"measurement": [

"used\_percent",

"used"

],

"metrics\_collection\_interval": 60,

"resources": [

"/"

]

}

}

}

}

Notes:

* namespace: "CWAgent" is the default namespace CloudWatch Agent uses.
* resources:[ "/"] tells it to monitor root filesystem only.
* used\_percent yields the disk\_used\_percent metric in CloudWatch.

Many 3rd-party or vendor packages (like Amazon’s CloudWatch Agent, Google Chrome, etc.) install into /opt instead of /usr/bin.  
But CloudWatch Agent isn’t included in the PATH by default, because it is in /opt/aws/amazon-cloudwatch-agent/etc/

So running amazon-cloudwatch-agent directly won’t work unless:

**Temporary test Option 1:** You specify the full path when using commands,

/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -v

**Temporary test Option 2:** Or you update PATH in your shell/profile:

export PATH=$PATH:/opt/aws/amazon-cloudwatch-agent/bin

**⚠️ But this only lasts until you log out or reboot – as it’s a temporary method.**

**-------------------------------  
✅ Make it permanent**

**For all users (system-wide)**  
Edit /etc/profile or create a new script under /etc/profile.d/:

sudo nano /etc/profile.d/cloudwatch-agent.sh

Put this inside:

export PATH=$PATH:/opt/aws/amazon-cloudwatch-agent/bin

Save + exit. This will apply to all users after re-login.

amazon-cloudwatch-agent –version #checking the version

**Verify the metric shows up (CloudWatch Console)**

* Console → CloudWatch → Metrics → CWAgent → Per-Instance Metrics (or browse the namespace).
* Look for disk\_used\_percent with a dimension InstanceId = i-... and path = /. (It may take ~1–2 minutes to appear.)

What is a **metric** in AWS?

A **metric** is a **time-ordered set of data points** that measure some aspect of your AWS resources or applications.

Each metric belongs to a **namespace** (like AWS/EC2 or CWAgent) and can have **dimensions** (key-value pairs like InstanceId=i-0123456789abcdef0).

There are two types of metrics:

| **Type** | **Description** | **Example** |
| --- | --- | --- |
| **AWS-provided metrics** | Automatically collected by AWS services | CPUUtilization for EC2 |
| **Custom metrics** | Sent by you (or an agent) using **CloudWatch PutMetricData** API | disk\_used\_percent from CloudWatch Agent |

**How to add a metric**

There are three main ways:

**🔹 Option A: Use AWS CloudWatch Agent**

1. Install CloudWatch Agent on EC2 (you already did).
2. Create a config JSON specifying metrics to collect:

{

"metrics": {

"namespace": "CWAgent",

"metrics\_collected": {

"disk": {

"measurement": ["used\_percent"],

"metrics\_collection\_interval": 60,

"resources": ["/"]

}}}}

Start the agent:

sudo systemctl start amazon-cloudwatch-agent

sudo systemctl enable amazon-cloudwatch-agent

* Agent will now **send disk\_used\_percent metrics to CloudWatch automatically**.

**C — Terraform: create SNS topic + CloudWatch alarm**

In here, terraform automates the manual creation of creating an alarm in aws.

Creating separate files inside a new folder,  
example: provider.tf, main.tf, variables.tf

**How to run Terraform**

From the folder:

terraform init

terraform plan \

-var="instance\_id=$INSTANCE\_ID" \

-var="alarm\_email=$ALARM\_EMAIL" \

-var="aws\_region=$AWS\_REGION"

terraform apply -auto-approve \

-var="instance\_id=$INSTANCE\_ID" \

-var="alarm\_email=$ALARM\_EMAIL" \

-var="aws\_region=$AWS\_REGION"  
fd  
  
After apply, Terraform prints the SNS topic ARN and alarm name.