**1. What is MFA**

AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of your user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication response from their AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources.

You can enable MFA for your AWS account and for individual IAM users you have created under your account. MFA can be also be used to control access to AWS service APIs.

After you've obtained a supported U2F security key, hardware device, or virtual MFA device, AWS does not charge any additional fees for using MFA.

**MFA Form Factors:**

* Virtual MFA Device
* Universal 2nd Factor (U2F) Security Key
* Hardware Key Fob MFA Device
* Hardware Display Card MFA Device
* SMS MFA Device (Preview)
* Hardware Key Fob
* MFA Device for
* AWS GovCloud (US)

**Virtual MFA Applications:**

Applications for your smartphone can be installed from the application store that is specific to your phone type. The following table lists some applications for different smartphone types.

**2. What is ec2 instance profile**

**Using Instance Profiles**

An instance profile is a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts.

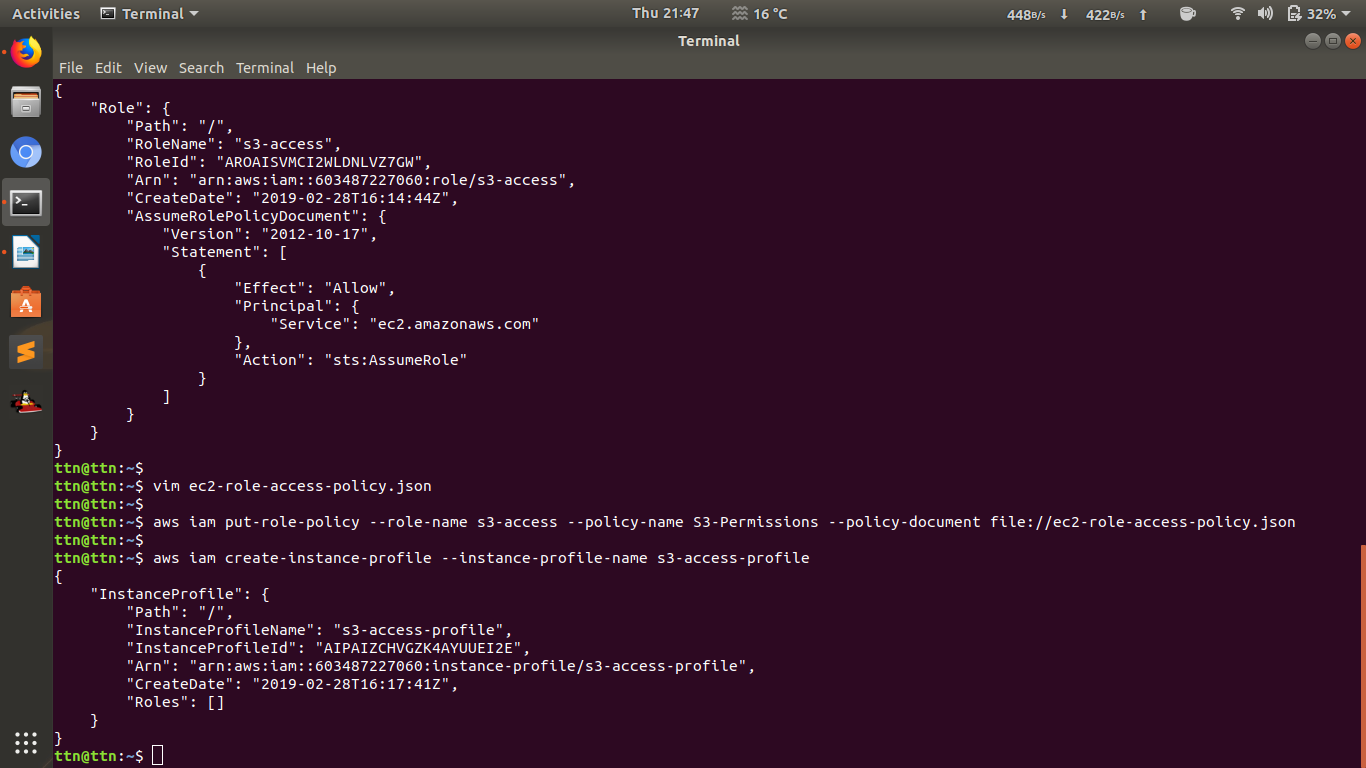
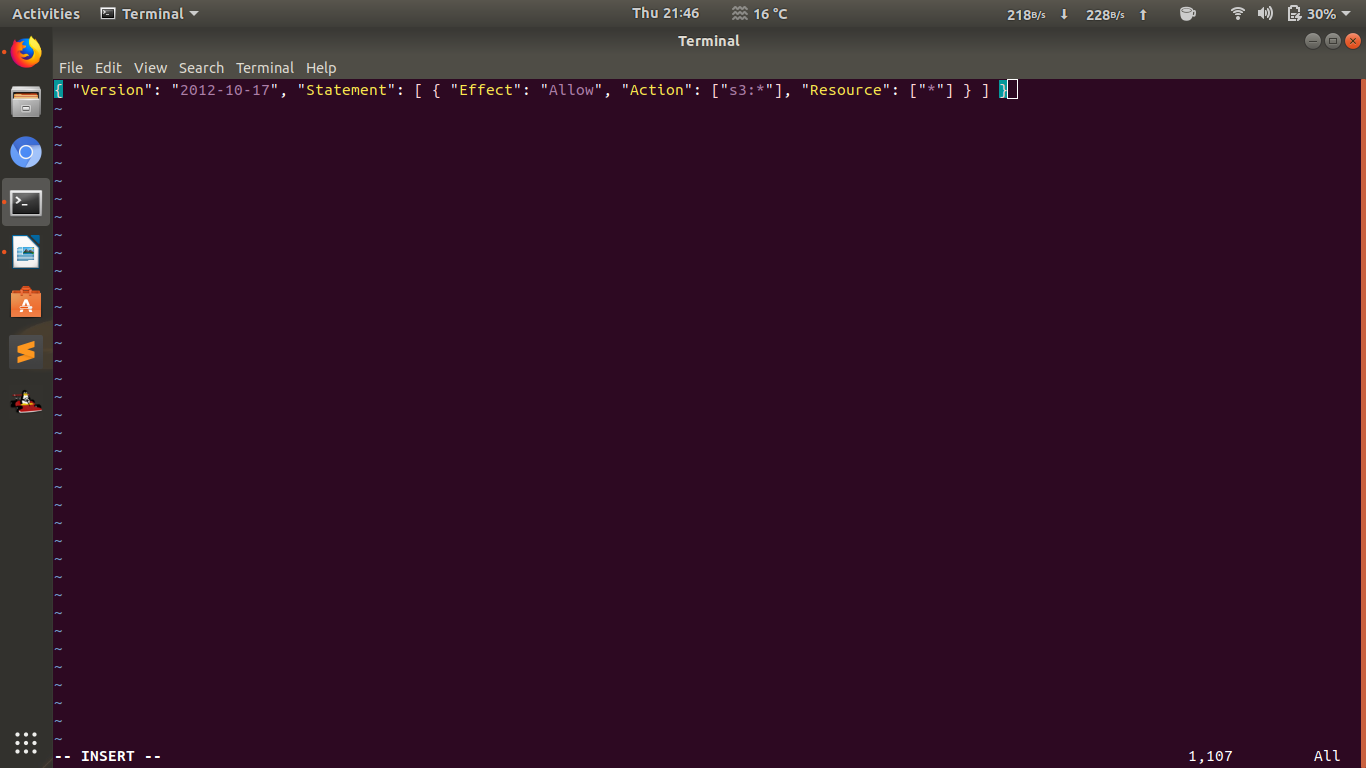
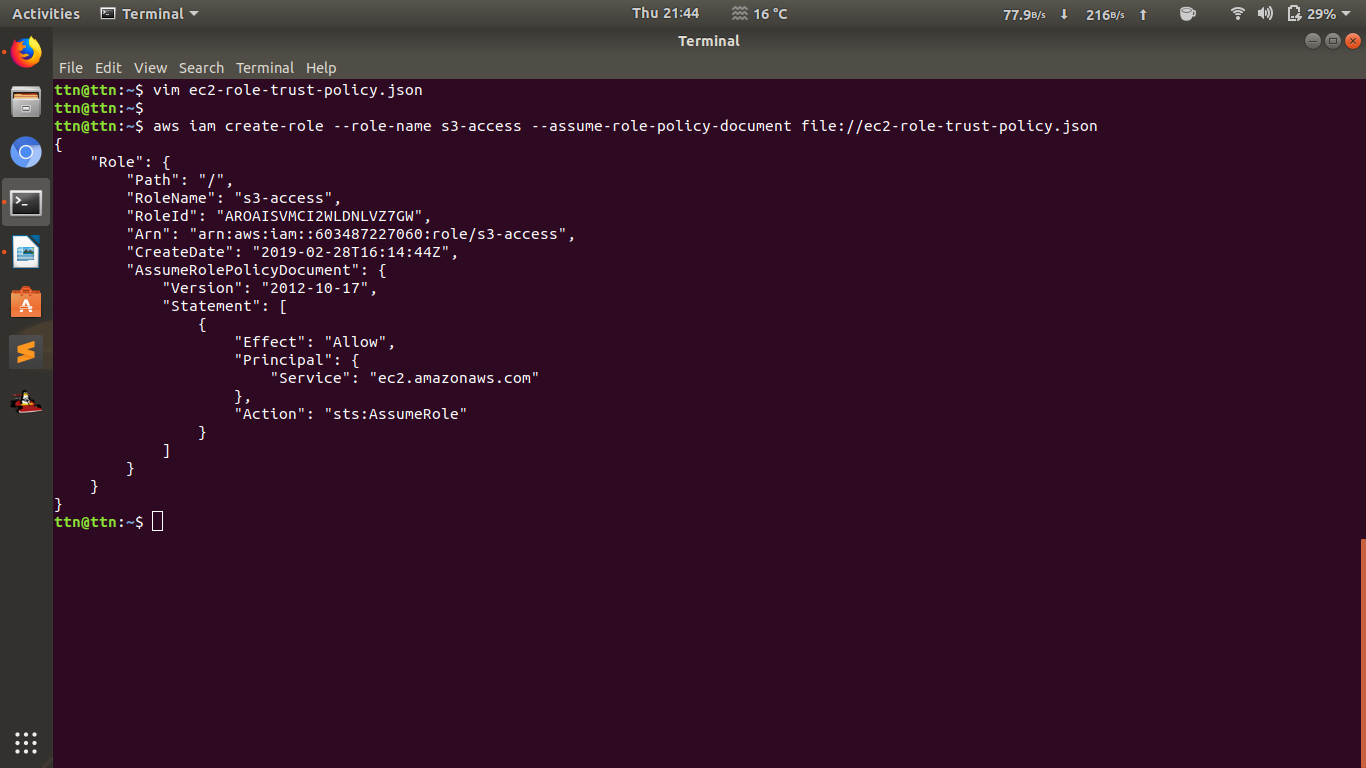
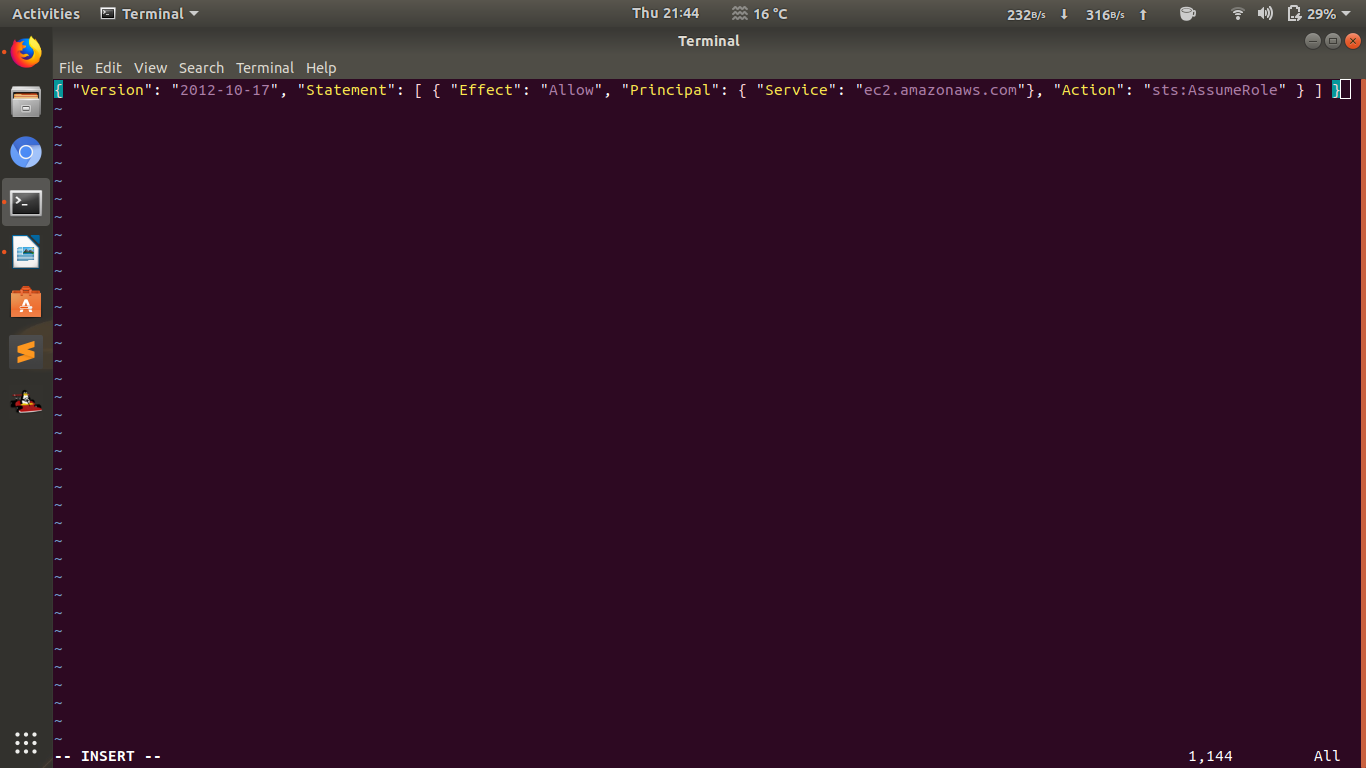
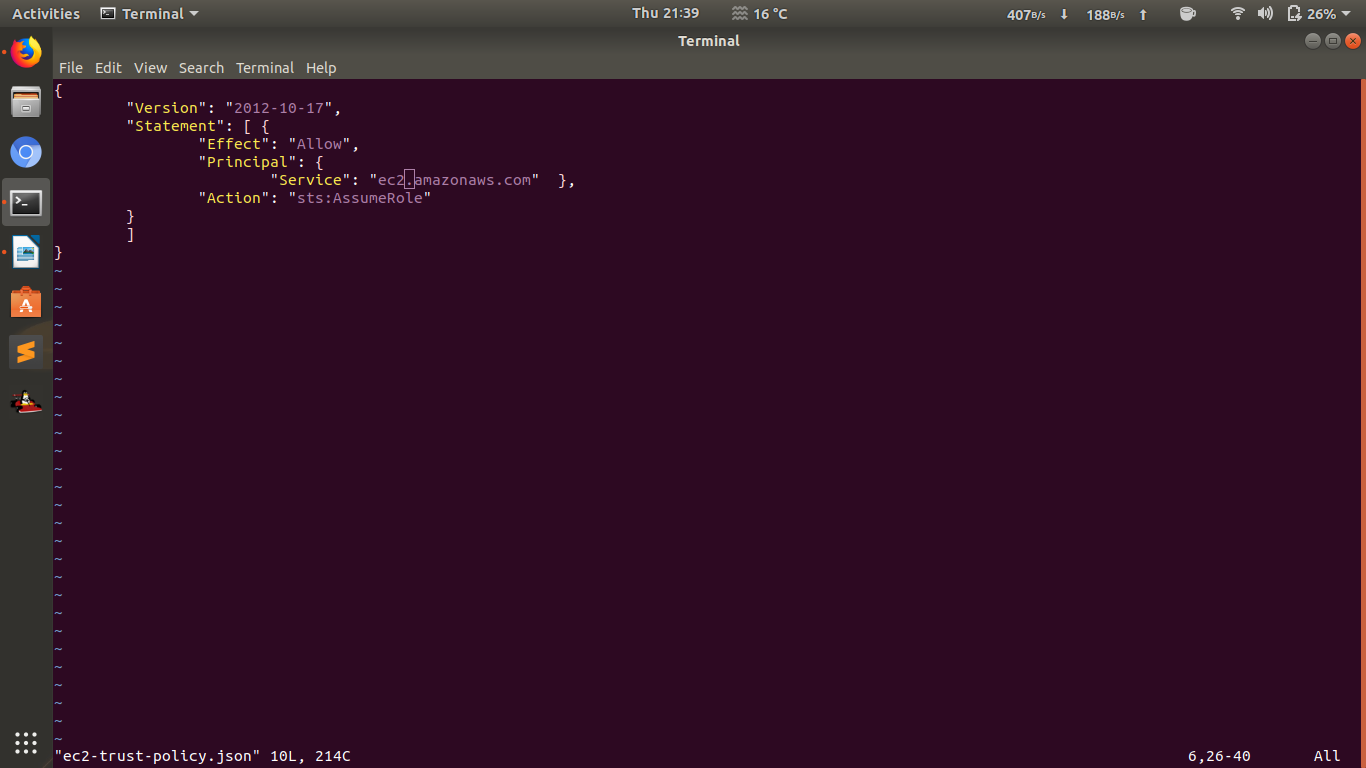
**Managing Instance Profiles (Console)**

If you use the AWS Management Console to create a role for Amazon EC2, the console automatically creates an instance profile and gives it the same name as the role. When you then use the Amazon EC2 console to launch an instance with an IAM role, you can select a role to associate with the instance. In the console, the list that's displayed is actually a list of instance profile names. The console does not create an instance profile for a role that is not associated with Amazon EC2.

**Managing Instance Profiles (AWS CLI or AWS API)**

If you manage your roles from the AWS CLI or the AWS API, you create roles and instance profiles as separate actions. Because roles and instance profiles can have different names, you must know the names of your instance profiles as well as the names of roles they contain. That way you can choose the correct instance profile when you launch an EC2 instance.

**3. Create a role from AWS CLI and which should have s3 full access permission and attach the role to an ec2 instance.**



**4. Create a policy which should have following things:**

**Should have a tag with name 'Owner'**

**'Owner' tag key should have some value.**

**The value must be equal to the username of the logged-in user.**

{

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

"Statement": [

{

"Sid": "Tag",

"Effect": "Deny",

"Action": "ec2:RunInstances",

"Resource": "arn:aws:ec2:\*:\*:instance/\*",

"Condition": {

"ForAllValues:StringNotEquals": {

"aws:TagKeys": "Owner"

},

"StringNotEquals": {

"aws:RequestTag/Owner": "\*"

}

}

},

{

"Sid": "Region",

"Effect": "Allow",

"Action": "ec2:\*",

"Resource": "arn:aws:ec2:\*:\*:instance/\*",

"Condition": {

"StringEquals": {

"ec2:ResourceTag/Owner": "${aws:username}"

}

}

}

]

}