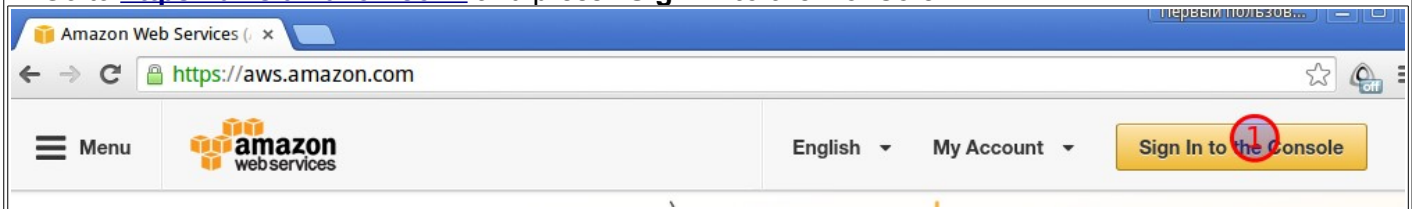
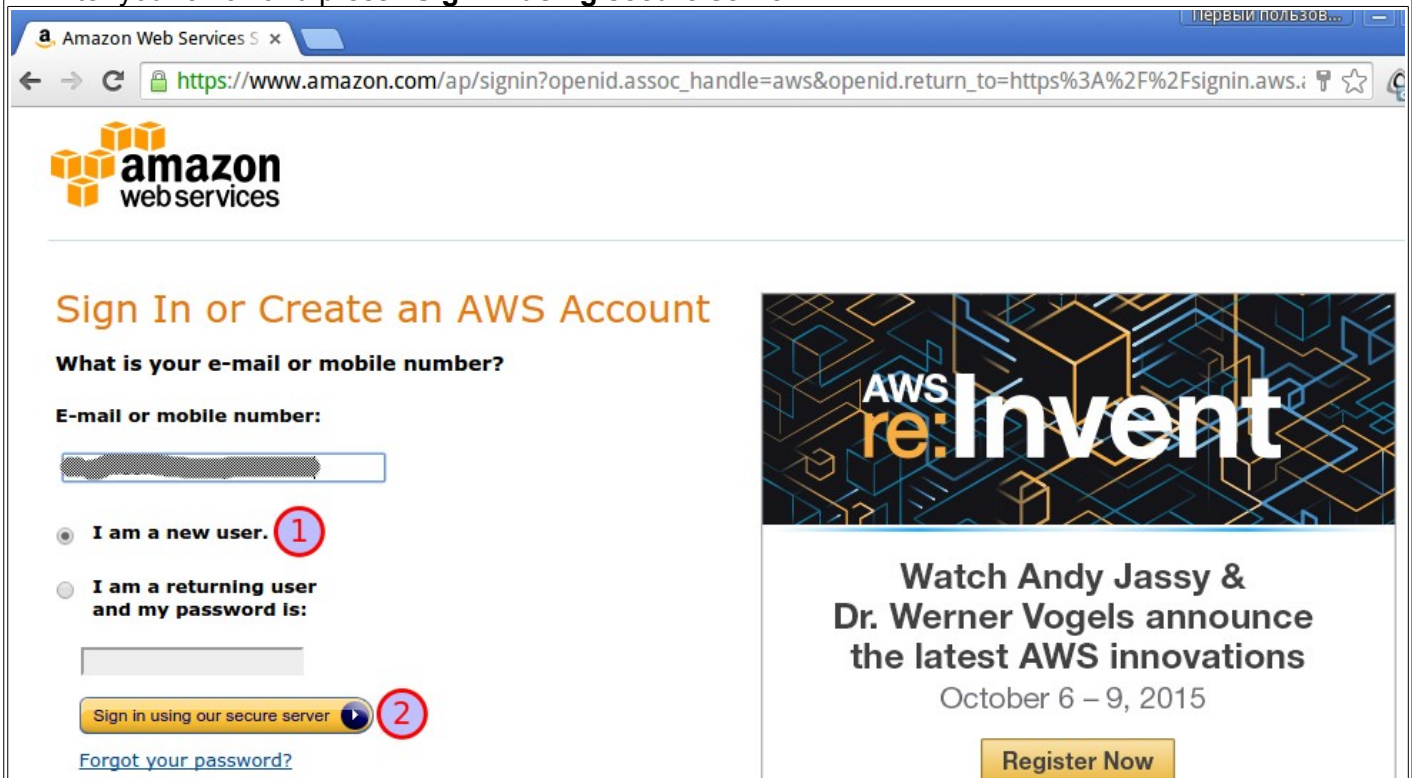


This document will guide you through install process of ubuntu server on aws and set it up for using with Minla LTE receiver board.

1. Go to <https://aws.amazon.com/> and press “Sign in to the Console”.



2. Enter your email and press “Sign in using secure server”.



3. On the next screens enter your registration information, contact information, payment information, pass identity verification and select **Basic (Free)** support plan.
Once you completed your registration press “Sign in to the Console” button again.

4. Login to console using your registration email and password.

Amazon Web Services

Sign In or Create an AWS Account

What is your e-mail or mobile number?

E-mail or mobile number:

☐ I am a new user.

☒ I am a returning user and my password is: 1

2

[Forgot your password?](#)

New AWS Accounts Include:

12 months of access to the AWS Free Tier

Amazon EC2: 750 hrs/month of Windows and Linux t2.micro instance usage

Amazon S3: 5GBs of Storage

Amazon RDS: 750 hrs/month of Micro DB Instance usage

Amazon DynamoDB: 25 GB of storage, up to 200 million requests/month

AWS Basic Support Features

Customer Service: 24x7x365

Support Forums

Documentation, White Papers, and Best Practice Guides

Visit aws.amazon.com/free for full offer terms.

5. Now we are in aws console. Select a region that is closest to you geographically. Then press on **EC2**.

AWS Management Console

https://eu-central-1.console.aws.amazon.com/console/home?region=eu-central-1

AWS Services Edit

Mike Frankfurt Support

Amazon Web Services

Compute

EC2 Virtual Servers in the Cloud 2

EC2 Container Service Run and Manage Docker Containers

Developer Tools

CodeCommit Store Code in Private Git Repositories

CodeDeploy Automate Code Deployments

Mobile Services

Cognito User Identity and App Data Synchronization

Device Farm Test Android, Fire OS, and iOS apps

Resource Groups

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

6. In EC2 Dashboard press **Launch Instance**.

Resources

You are using the following Amazon EC2 resources in the EU Central (Frankfurt) region:

0 Running Instances	0 Elastic IPs
0 Volumes	0 Snapshots
0 Key Pairs	0 Load Balancers
0 Placement Groups	1 Security Groups


Easily deploy and operate applications - use Chef recipes, manage SSH users, and more. [Try OpsWorks now.](#)

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

1

7. Select Ubuntu Server.


Ubuntu
Free tier eligible

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-accff2b1
Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root device type: ebs Virtualization type: hvm

Select **1**
64-bit

8. Select **t2.micro** instance type and press button “Next: Configure Instance Details”

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate

9. Press “Next” until step 6 (“Configure Security Group”). This is your server firewall settings. For now just select “Select an existing security group” and use the default one. **Make sure you review the details of this security group later to confirm that all traffic to and from any source is allowed.**

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☐ Create a new security group

☒ Select an existing security group **1**

Security Group ID	Name	Description	Actions
<input checked="" type="checkbox"/> sg-60ff9b09 2	default	default VPC security group	Copy to new



Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

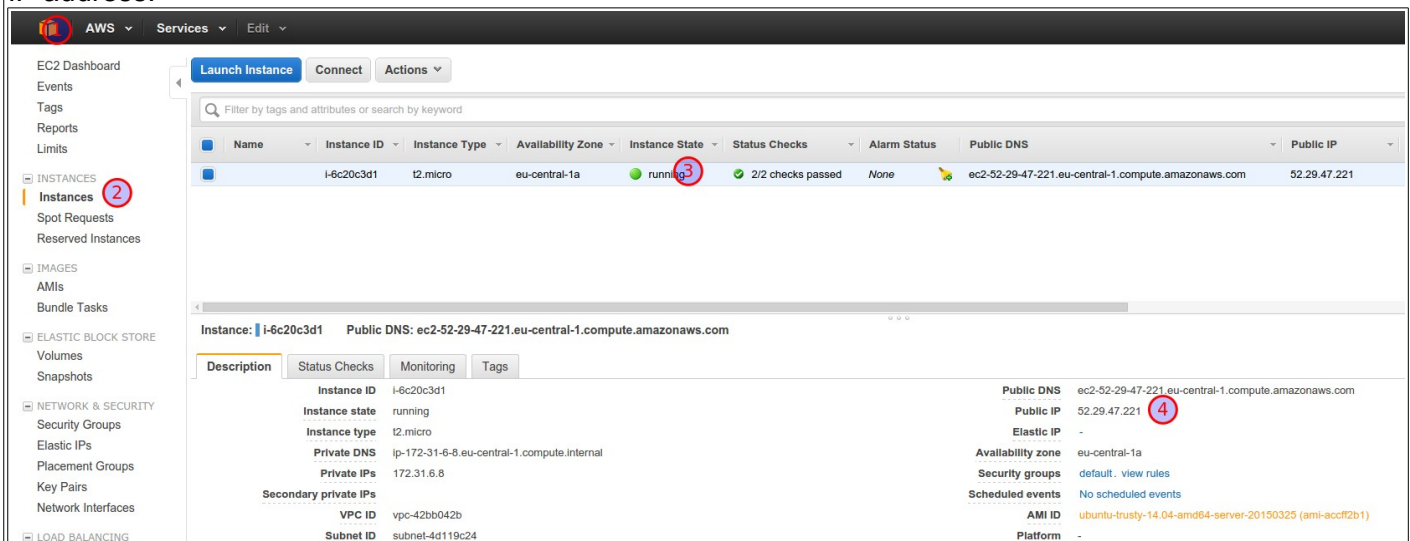
10. Proceed with “Review and Launch” and “Launch”.

At some step you will be asked for keypair. You need to create a new one or use an existing if you already created one before. It's important because without keypair you will loose access to the server.

So for now just create a new one. Remember to download and save it in known place on your hard drive. Keep it secure.

When completed you can launch your new instance.

11. Next go back to EC2 control panel, select “Instances”, check that state is “running” and note your Public IP address.



The screenshot shows the AWS Management Console for EC2. The left sidebar has a red circle around the 'Instances' link. The main content area shows a table of instances. The first instance, 'i-6c20c3d1', is in the 'running' state. The 'Public IP' column shows '52.29.47.221', which is circled in red. Below the table, the details for instance 'i-6c20c3d1' are shown, including its Public DNS, Public IP, Availability zone, Security groups, and AMI ID.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
	i-6c20c3d1	t2.micro	eu-central-1a	running	2/2 checks passed	None	ec2-52-29-47-221.eu-central-1.compute.amazonaws.com	52.29.47.221

Instance: i-6c20c3d1 Public DNS: ec2-52-29-47-221.eu-central-1.compute.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID: i-6c20c3d1	Instance state: running	Instance type: t2.micro	Private DNS: ip-172-31-6-8.eu-central-1.compute.internal
Private DNS: 172.31.6.8	Secondary private IPs	VPC ID: vpc-42bb042b	Subnet ID: subnet-4d119c24

Public DNS: ec2-52-29-47-221.eu-central-1.compute.amazonaws.com
Public IP: 52.29.47.221
Elastic IP: -
Availability zone: eu-central-1a
Security groups: default, view rules
Scheduled events: No scheduled events
AMI ID: ubuntu-trusty-14.04-amd64-server-20150325 (ami-acff2b1)
Platform: -

12. At this step we finished with AWS web site.
Next we will login to server via SSH and configure it to run web server and node.js.
Open SSH terminal (in my case I will use Ubuntu linux on my desktop).

```
Terminal - g@g-I
File Edit View Terminal Tabs Help
g@g-Inspiron-5737:~$ ssh -i /home/g/test.pem ubuntu@52.29.47.221
```

If login is successful you should see following:

```
Terminal - ubuntu@ip-
File Edit View Terminal Tabs Help

System information as of Sun Oct 25 14:31:20 UTC 2015

System load: 0.0          Processes:           97
Usage of /:  9.8% of 7.74GB Users logged in:       0
Memory usage: 5%          IP address for eth0: 172.31.6.8
Swap usage:  0%

Graph this data and manage this system at:
https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Last login: Sun Oct 25 14:31:21 2015 from 91.205.144.229
ubuntu@ip-172-31-6-8:~$
```

Next just run following commands in terminal:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install apache2
curl --silent --location https://deb.nodesource.com/setup_4.x | sudo bash -
sudo apt-get install --yes nodejs
cd /var/www/html/
```

```
sudo npm install forever -g
sudo npm install ws --save
sudo chmod 0777 /var/www/html
```

Copy server *.js file and control panel *.html file to /var/www/html by any sftp client.

To run server js script you need to do:

```
screen
forever /var/www/html/server.js
```

Then press CTRL+A+D and close ssh terminal.

You can test if your web server works by accessing your AWS ip address/qc.html in your browser.