

Ignat Sparnou
(*F. Skaryna GSU, Gomel*)

DEPLOY A NODE.JS APPLICATION USING AWS EC2 SERVICE

Рассмотрены вопросы развертывания приложения Node.JS с помощью службы AWS EC2.

EC2 (elastic compute cloud) is a service that provides computing power by providing virtual / physical machines with pre-imaged operating systems (e.g. Ubuntu, Debian, Windows).

AWS EC2 uses AMI (Amazon machine image), which is the image of the prepared operating system that the EC2 instance will use. The clos-

best analogy is a virtual disk with an already installed system, for example, in Virtual Box.

Immediately after registering for the service, you are taken to the AWS EC2 console. There you can select EC2 and the type of car on which everything will be deployed.

The application will be deployed in two stages. To deploy, go to the created machine, make an update, install Node.js, clone the code from the repository using the git clone command, install npm modules.

The next step is to create your own AMI image. Based on this AMI image, you can make a copy of the backend and deploy it in two clicks. To create, go to "Image" – "Create Image". Next, you should specify the minimum size of the hard disk. We are talking about the size of the snapshot, that is, in fact, a snapshot will be taken from the hard disk on which everything is installed. We can say that AMI is a virtual entity consisting of image settings and a snapshot with an installed OS (in our case, an OS with Node.js installed and our application).

It will take about five minutes to create a snapshot, during which the car will be in the hold status (the "fatter" your image, the longer the hold will take).

After creating the image, you can replicate it and get the same copy of the machine in just a few clicks. And connect to the machines not individually, but through load balancers. There are several balancers to choose from, which differ in balancing algorithms. Setting up the selected balancer will not cause difficulties, although there are nuances that are important to remember.