

MODERN PROBLEMS OF DEPLOYMENT AND MAINTENANCE OF HIGH-LOADED APPLICATIONS

MIKITA TORAPAU, ARKADII OSKIN

Polotsk State University, Belarus

This article is about recent problems of deployment and managing high-loaded systems and applications and their solutions.

Introduction. Iterative method has become the most popular and the most efficient method of application development, it means that deployment stage has become a significant part of a life cycle for most applications. Of course, this stage should be automated as much as possible, because it takes a lot of time to write your code, deploy it, configure environment and launch your application.

Continuous integration and continuous delivery have become almost the requirement for developing high-quality applications. Modern market offers a list of products for realization of these concepts, and it's important to choose better solution.

Generally, development process consists of three parts: development, testing and deployment. In case of high-loaded systems deployment can be the most important stage, because applications under load may cause various problems and it's very important to setup environment in the right way.

Amazon Web Services provides a huge amount of tools for automating deployments and releases, for testing your applications and systems; it makes the development process faster and easier. AWS CodePipeline offers a smart automated solution for an application build, test, and release process, allowing fast and reliable application updates.

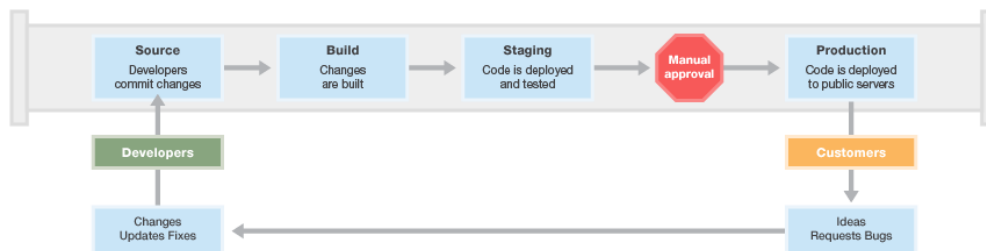


Fig. 1. AWS CodePipeline release process

Using AWS CodePipeline, developers can define a full release process for building code, deploying to pre-production environments, testing the application, and releasing to production – all from one platform. It allows performing live load tests of any size. AWS CodePipeline offers advanced scripting capabilities, a flexible SaaS platform for easy test executions, scheduling, automation and results analysis, GUI-supported scripting with no programming required, full API, and integration with APM tools including AppDynamics and New Relic [1].

Another problem of deployment is how to setup your environment. It can be achieved in different ways, but the smartest and most flexible solution is "Infrastructure as Code". Infrastructure as Code is the process of managing and provisioning computing infrastructure (processes, services databases, virtual servers, etc.) and their configuration through machine-processable definition files, rather than physical hardware configuration or the use of interactive configuration tools [2].

AWS CloudFormation enables you to create and provision AWS infrastructure deployments predictably and repeatedly. It helps you leverage AWS products such as Amazon EC2, Amazon Elastic Block Store, Amazon SNS, Elastic Load Balancing, and Auto Scaling to build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS infrastructure [3]. AWS CloudFormation enables you to use a template file to create and delete a collection of resources together as a single unit. Also AWS CloudFormation has built-in tools for monitoring your environments. It allows using various metrics to check your instances, like CPU utilization, memory utilization, application load, incoming and outgoing requests, disk utilization and other.

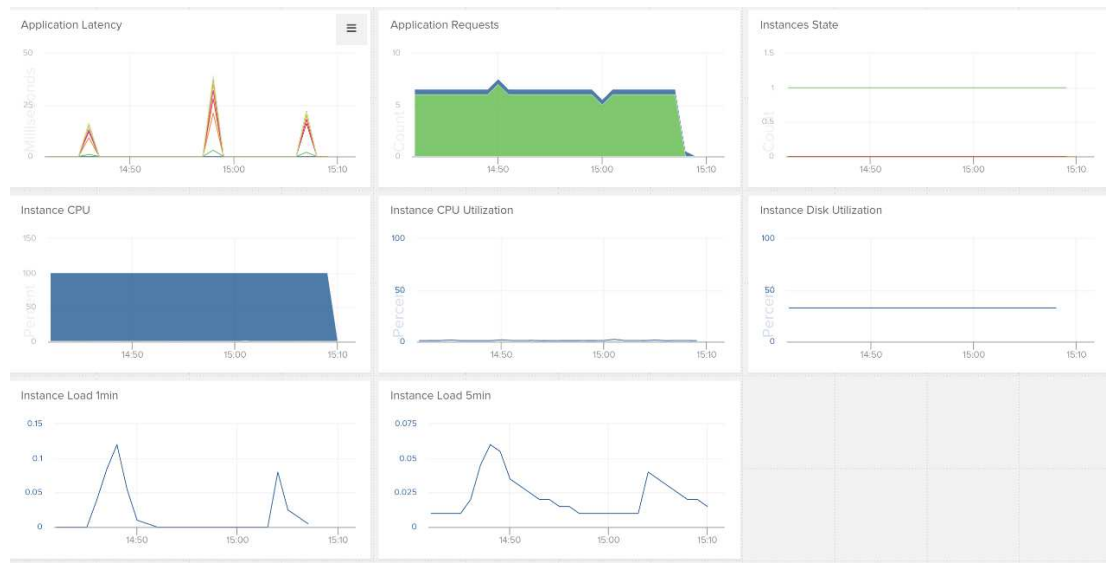


Fig. 2. AWS Cloudformation metrics

Cloudformation works great with microservice architecture, because in case of adding a new microservice all you should do is to define a new container in a template for your service rather than changing the whole structure of your environment.

When an application is run, it's important to know about any abnormal effects in your environment. For example, if your environment has a high CPU utilization or too many requests are sent to your server, you should be notified to be able to prevent any problems as soon as possible. Amazon CloudWatch provides a reliable, scalable, and flexible monitoring solution that you can start using within minutes. You no longer need to set up, manage, and scale your own monitoring systems and infrastructure. You can define your custom alarm, set an alarm rule and get a notification when something happens. Besides, AWS CloudWatch allows storing logs from Amazon EC2 instances, load balancers, or other sources.

The configuration of the environment for high-loaded applications is different from other applications. Of course, each application has its own requirements for building environment, but there are some general principles:

1. Automated backups.

If an application stores any important data in databases, all data should be backed up to be able to restore it.

2. Load balancing.

Load balancing improves the distribution of workloads across multiple computing resources. It helps to minimize response time, optimize your instance and avoid overload of any single resource. With the load balancer you have just one public DNS and many instances.

3. Auto scaling.

Auto Scaling is a web service designed to launch or terminate Amazon EC2 instances automatically based on user-defined policies, schedules, and health checks. Auto scaling works perfectly with the load balancing and alarms. For example, when CPU utilization increases greatly, a new instance will be added to a load balancer, and when there is no load on the server, the instance will be terminated.

4. Using private virtual network for environments.

A system or an application can be encapsulated with the help of a private virtual network. All resources can be placed into this network, and only needed port will be exposed. It's a good solution for improving application security.

5. Using proxies and multiple instances in different regions.

Sometimes requests have long response time because of a long distance between a client and a server. The solution is to place environments in different regions.

Conclusion. Amazon Web Services is the best solution for deploying and managing a large high-loaded system as it has all tools for deploying, managing and monitoring your environment, testing, building and releasing your application. With the help of AWS all general principles of deployment of heavy loaded systems can be easily achieved. AWS allows developers to create reusable solutions for different projects. AWS SDK provides a rich API for deploying instances, managing environments, testing applications, building snapshots using various

programming languages such as Java, .NET, Javascript, PHP and others. All resources are created in a cloud, so you have to pay only for what you need.

REFERENCES

1. AWSCodePipeline – AWS Documentation [Электронныйресурс] / AWSCodePipeline. – Режим доступа: <https://aws.amazon.com/documentation/codepipeline/>. – Дата доступа: 20.01.2017.
2. Wikipedia [Электронныйресурс] / Infrastructure as Code. – Режим доступа: http://en.wikipedia.org/wiki/Infrastructure_as_Code/. – Дата доступа: 20.01.2017.
3. AWSCloudformation–AWSDocumentation [Электронныйресурс] / AWS Cloudformation. – Режим доступа: <http://aws.amazon.com/documentation/cloudformation/http://habrahabr.ru/company/redmadrobot/blog/252773/>. – Дата доступа: 20.01.2017.