



Interacting with AWS to turn system diagrams into reality

By Michael Wittig

In this article, you will get an overview of tools that are available to make calls to the AWS API.

Generally speaking you can host any application on AWS. If you interact with AWS, you will always make calls to the API. The API is the interface between you and AWS as figure 1 demonstrates.

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/wittig/>

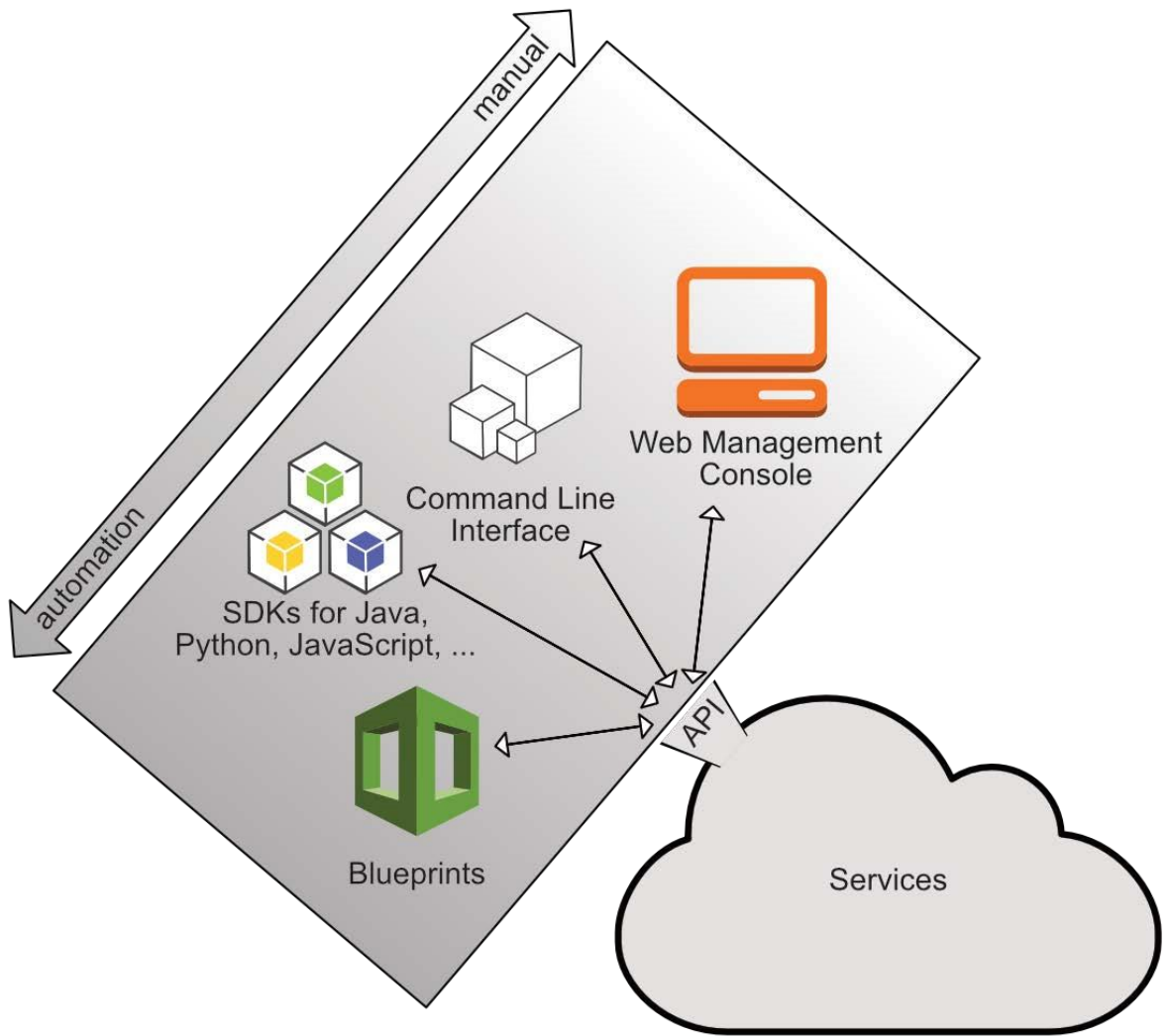


Figure 1 Tools to interact with the AWS API

You will now get an overview of tools that are available to make calls to the AWS API. Please compare the following tools in their ability to automate your daily tasks.

Web Management Console

You can interact with AWS by using the web-based Management Console. You can manually control AWS with this convenient GUI that runs in every modern web browser (Chrome, Firefox, Safari >= 5, IE >= 9). Figure 2 shows what the Management Console looks like.

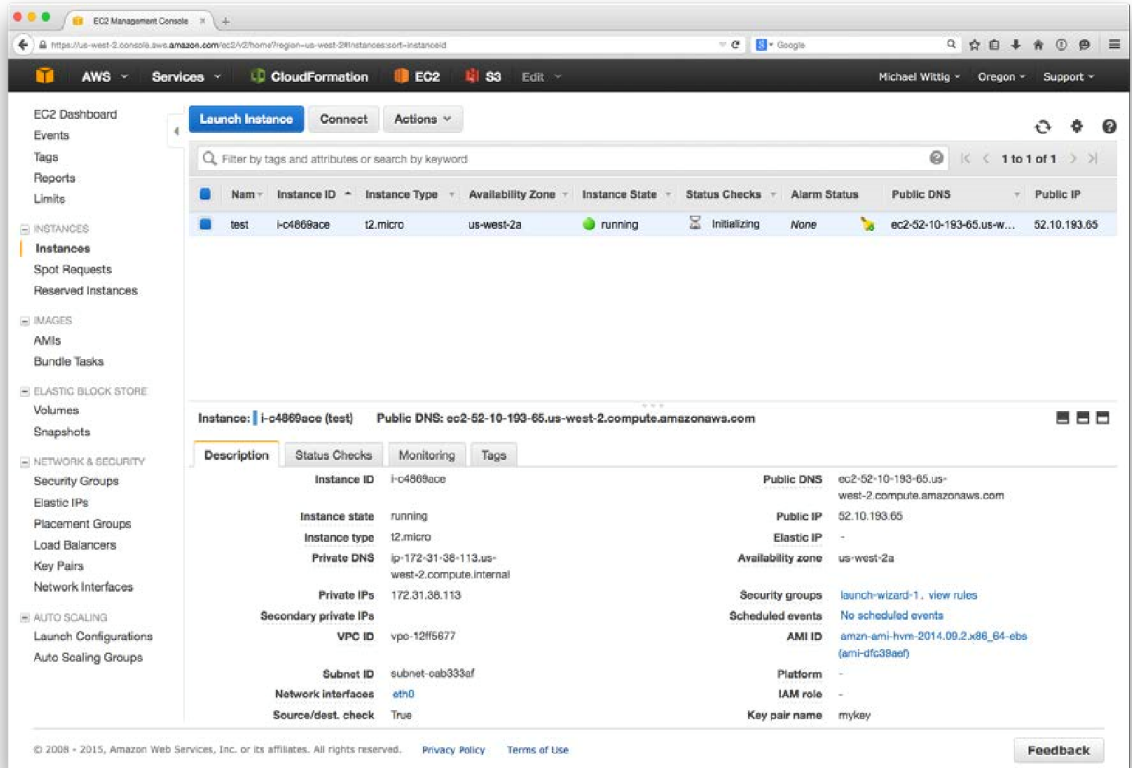


Figure 2 Web Management Console

If you're doing some experiments with AWS, the Management Console is the best place to start. It helps you to gain an overview of the different services and achieve first success quickly. The Management Console is also a good way to set up a cloud infrastructure for development and testing.

(You can read an exploration of the Management Console in section 1.7 of my book, [Amazon Web Services in Action](#).)

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://www.manning.com/wittig/>

Command Line Interface

You can start a virtual server, create storage, and send mails from the command line. With the Command Line Interface (CLI) you can control everything in AWS. Figure 3 shows the CLI in action.



```
michael — bash — 92x39
Last login: Fri Feb 20 09:32:45 on ttys000
mwittig:~ michael$ aws cloudwatch list-metrics --namespace "AWS/EC2" --max-items 3
{
  "Metrics": [
    {
      "Namespace": "AWS/EC2",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-ed62dc0b"
        }
      ],
      "MetricName": "StatusCheckFailed_Instance"
    },
    {
      "Namespace": "AWS/EC2",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-ed62dc0b"
        }
      ],
      "MetricName": "StatusCheckFailed"
    },
    {
      "Namespace": "AWS/EC2",
      "Dimensions": [
        {
          "Name": "InstanceId",
          "Value": "i-0a02beec"
        }
      ],
      "MetricName": "CPUUtilization"
    }
  ],
  "NextToken": "None___3"
}
mwittig:~ michael$
```

Figure 3 Command Line Interface

The CLI is typically used to automate tasks on AWS. If you want to automate parts of your infrastructure with the help of a continuous integration server, like Jenkins, the CLI is the right tool for the job. The CLI offers a convenient way for accessing the API and combining multiple calls into a script.

You can even start to automate your infrastructure with scripts by chaining multiple CLI calls together. The CLI is available for Windows, Mac and Linux. There is also a PowerShell version available.

SDKs

Sometimes you need to call AWS from within your application. With SDKs you can use your favorite programming language to integrate AWS into your application logic. AWS provides SDKs for:

- Android
- Browser (JavaScript)
- iOS
- Java
- .Net
- Node.js (JavaScript)
- PHP
- Python
- Ruby

SDKs are typically used to integrate AWS services into applications. If you are doing software development and want to integrate AWS services like a NoSQL database or a push notification service, SDKs are the right choice for the job.

Some services like queues or topics must be used with a SDK in your application.

Blueprints

A blueprint is a description of your system containing all services and dependencies. The blueprint does not say anything about the necessary steps or their order for achieving the described system. Figure 4 illustrates how a blueprint is transferred into a running system.

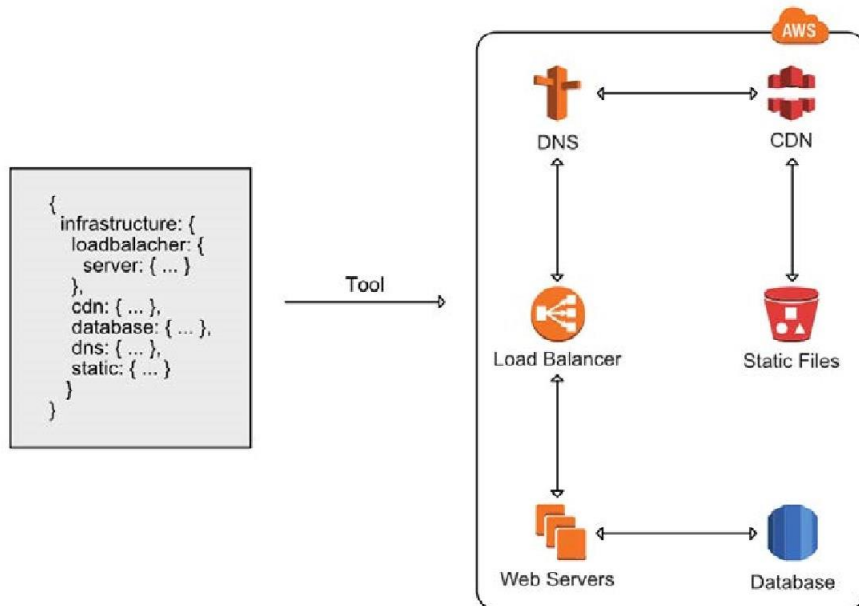


Figure 4 Infrastructure automation with blueprints

You use blueprints if you have to control many or complex environments. Blueprints help you to automate the configuration of infrastructure in the cloud. You can use blueprints to set up virtual networks and launch different servers into that network, for example.

A blueprint removes a lot of burden from you because you no longer need to worry about dependencies during system creation. A blueprint automates the whole process of system creation.