



UMUC

University of Maryland University College

CC615x

# Cloud Infrastructure

Week 5: Infrastructure Services and Cloud Architectures Part 1

# Objectives

In this week's lesson, you will learn:

IaaS Application Programming Interfaces (APIs)	Auxiliary Services in IaaS Environments
<ul style="list-style-type: none"><li>• What are APIs?</li><li>• IaaS APIs</li><li>• Infrastructure as Code</li></ul>	<ul style="list-style-type: none"><li>• IaaS Security</li><li>• Monitoring Cloud Deployments</li><li>• Database Services</li></ul>



## Objectives, Cont.

In this week's lesson, you will also learn:

### Typical Cloud IaaS Architectures

- Single Server
- Single Server with Services
- Application with Security Zones and On-Premise IT Connectivity
- Load Balancing, Auto-Scaling, Database Replication
- Fault Tolerance and High Availability
- Fast Delivery of Static Content – Object Storage, Content Delivery Network
- Architecture Components Can Be Mixed and Matched



**UMUC**

University of Maryland University College

# IaaS Application Programming Interfaces (APIs)



# What are APIs?

- The majority of applications users are familiar with are intended to be used by humans and are equipped with human-friendly, graphical user interfaces (GUIs). Cloud vendors also allow customers to interact with their offerings via web portals with GUIs that allow manual provisioning and controlling of cloud offerings.
- These days, computer programs tend to talk more to each other than to humans and the language they use is based on Application Programming Interfaces (APIs). The business domain of a computer system is modeled by creating its data model, defining system entities, and their relationships. The functionality of a software system is then made available as a library of function calls, taking data elements and entities as input arguments and returning processed results.



# What are APIs, Cont.

- The fact that virtual cloud infrastructure elements; including compute, storage, and network; are defined and controlled by software makes them particularly suitable for programmatic interactions.
- A very popular API style is REST (Representational State Transfer). It assumes that client-servers, stateless communications, no interaction state is persisted between calls, and all resources needed are passed as arguments or returned as results. The protocol of the Web, HTTP, fits this style particularly well with its simple GET, POST, PUT and DELETE operations.
- For more information on REST, see (links can be found in course or PowerPoint):  
[https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)



## What are APIs, Cont.

- The REST API style fits the cloud deployment models because it works well with horizontal scaling. Most cloud vendors make their IaaS offering APIs available in the form of REST libraries.
- In addition, they offer development kits for popular Web programming languages, allowing the calls to be made from programs written in these languages, as well as Command Line Interface (CLI) packages which help write command scripts. This fits well with another cloud application programming style: Polyglot (multi-language) programming.
- The importance of APIs is leading to the creation of the “API economy”: software vendors defining, running, and selling API access to their systems to be accessed by other software systems over Internet.



# laaS APIs

- Frequently, cloud laaS API calls are used when implementing cloud providers' web portals. Therefore, customers can invoke most of the same functionality using APIs as when using the cloud provider's web portal. They can provision resources, manage and monitor them, retrieve billing information, etc.
- The API call arguments have to include the authentication information to make sure the call is authorized. For example, below is an outline of a REST call which, when put into a Web browser, will start an instance with the given Id in AWS (links can be found in course or PowerPoint):

<https://ec2.amazonaws.com/?Action=StartInstances&InstanceId.1=i-10a64379&AUTHPARAMS>



## IaaS APIs, Cont.

- Cloud APIs make it possible to write programs that define the dynamic virtual environment in which the deployed software will run. This opens an entirely new set of possibilities in deploying and developing software, leading to fully automated DevOps and self-configurable software systems.
- See API links in course or on PowerPoint:

AWS: <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html>

Azure: <https://docs.microsoft.com/en-us/rest/api/>



# Infrastructure as Code

- To help create and control cloud IaaS resources programmatically, most vendors provide a way of textually defining resource configurations so that they are both human- and program-readable. This capability is called Infrastructure as Code (IaC).
- Compute, storage, and network resource configurations can be described as templates utilizing notations such as Java Simple Object Notation (JSON) or “Yaml Ain’t Markup Language” (YAML). These notations allow for textual representation of nested structures using name value pairs. Templates may be parameterized so that they can be reusable.
- Amazon AWS has a template feature called CloudFormation. For more information and examples, see (links can be found in course or PowerPoint):  
<http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-what-is-concepts.html>
- IaC allows for describing complex deployment structures as templates and automatically deploying them via command line interfaces or APIs. It is a very powerful feature, providing for ultimate cloud automation.





**UMUC**

University of Maryland University College

**Continue to Part 2 of the Video Lecture**