

## **CLOUD COMPUTING**

Prof. Raffaele Montella

### **Obiettivi formativi**

Il corso è un'introduzione alla programmazione di applicazioni in ambiente cloud computing con riferimenti anche allo high performance computing, al calcolo distribuito, alla computazione al bordo e all'internet delle cose. Il corso contiene un'introduzione alle principali tecnologie per cloud computing di tipo Infrastructure as a Service, Platform as a Service e Software as a Service. Una parte del corso è dedicata all'introduzione ai servizi offerti da Amazon Web Services e l'interfacciamento con single board computer e microcontrollori di tipo Arduino o similari. Nel corso delle lezioni sono fatti continui riferimenti alle tecnologie per lo sviluppo di applicazioni web, mobile e mobile ibride multiplatforma, alla gestione e rappresentazione dei dati spaziali e ai sistemi pervasivi distribuiti/indossabili.

Il corso contiene una introduzione al linguaggio Python, utilizzato per lo sviluppo di progetti software e hardware nelle attività (singole o di gruppo) in Laboratorio che sono parte integrante del corso. Il corso ha una impostazione tipicamente pratico/applicativa finalizzata alla risoluzione di problemi concreti e allo sviluppo di software effettivamente utilizzabile.

Conoscenza e capacità di comprensione: Lo studente deve dimostrare di conoscere e saper comprendere i fondamenti delle tecnologie direttamente o indirettamente collegate al cloud computing e dell'internet of things, con particolare riguardo alle applicazioni distribuite, le metodologie di sviluppo e di analisi di applicazioni basate su servizi web, il linguaggio di programmazione Python e il suo contesto applicativo, le metodologie e le tecniche dello sviluppo software in tale contesto.

Capacità applicative: Lo studente deve dimostrare di saper utilizzare la propria conoscenza acquisita per risolvere problemi computazionali come applicazioni distribuite su cloud che scalano dinamicamente da sviluppare in contesti concreti, per sviluppare soluzioni progettuali e analizzarle dal punto di vista dell'architettura e dell'usabilità, per usare in modo consapevole uno o più linguaggi di programmazione al fine di implementare un'applicazione web complessa. Tali capacità si estrinsecano anche in un uso approfondito e consapevole degli strumenti offerti dai differenti framework applicativi (sviluppo basato su piattaforme).

Autonomia di giudizio: Lo studente deve essere in grado di sapere valutare in maniera autonoma i risultati ottenuti dalla applicazione sviluppata sia in termini progettuale che dell'implementazione software valutandone l'efficacia e l'efficienza.

Abilità comunicative: Lo studente deve essere in grado di redigere una relazione di presentazione di un progetto e di documentare la sua implementazione, anche lavorando in gruppo, servendosi di strumenti avanzati di scrittura/documentazione con particolare riferimento all'uso degli strumenti di produttività per sviluppatori come sistemi per la redazione condivisa di documenti (Google Drive), di collaborazione remota (Slack), di gestione della produzione (Trello), di gestione dei repository online (GitHub), di infrastrutture cloud come servizio (Amazon AWS) usando correttamente la terminologia delle tecnologie web anche in lingua inglese.

Capacità di apprendimento: Lo studente deve essere in grado di aggiornarsi e approfondire in modo autonomo argomenti e applicazioni specifiche di tecnologie cloud, anche accedendo a banche dati, repository on-line di documenti, software e librerie applicative e altre modalità messe a disposizione dalla rete.

### **Contenuti**

Gestione dei dati nella cloud  
Calcolo utilizzando risorse di cloud computing  
Il cloud computing come una piattaforma  
Costruire una propria cloud privata  
La sicurezza nel cloud e argomenti ancillari

### **Lingua Insegnamento**

Italiano. Se presente almeno uno studente Erasmus, il corso sarà tenuto in Inglese.

### **Programma dettagliato**

Lo spazio di memorizzazione come servizio  
Usare i servizi di memorizzazione

Un database distribuito: CosmosDB  
Risorse di calcolo come servizio  
Usare e gestire le macchine virtuali  
Usare e gestire i container  
Applicativi di tipo scalabile  
Singularity: un sistema per applicazioni HPC  
Sistemi Serverless and funzioni come servizio  
Analisi dei dati nel cloud  
Fare streaming di dati nel cloud  
Machine learning in cloud  
Supplement: Autoencoders and Manifold Learning  
Il sistema Globus per la gestione dei dati in attività di ricerca  
Creare la propria cloud con Eucalyptus  
Creare la propria cloud con OpenStack  
Creare il proprio sistema Software as a Service  
Sicurezza e privacy  
Storia critica e sviluppi futuri.

### **Modalità di verifica dell'apprendimento**

L'obiettivo della procedura di verifica consiste nel quantificare il livello di raggiungimento degli obiettivi formativi precedentemente indicati.

La procedura di verifica è indicata precisamente nella piattaforma di e-learning del Dipartimento di Scienze e Tecnologie. In sintesi: a circa 1/3 del corso gli studenti sottomettono una proposta di progetto rafforzata da una presentazione composta da non più di 3 slide e della durata non superiore ai 3 minuti. La proposta può essere individuale o di un gruppo composto da non più di 3 studenti. La proposta viene emendata dal docente, rigettata o assegnata come progetto di esame. A circa 2/3 del corso è organizzata una presentazione dei progetti in cui ciascuno studente o gruppo di studenti illustra lo stato dell'arte del progetto e discute con docenti e altri studenti di potenzialità e debolezze dello stesso. A fine corso è organizzata una presentazione finale che non ha valore di esame ma momento comunicativo del lavoro svolto. Il progetto deve essere rilasciato in open source preferibilmente con licenza Apache 2.0 su di un repository pubblico come, ad esempio, GitHub almeno 2 giorni prima della data di esame e deve essere funzionante online almeno fino alla conclusione dell'esame. La procedura di verifica consiste in un esame orale individuale (60% del voto), la presentazione della parte di progetto di propria competenza (20% del voto), dimostrazione del progetto (20% del voto). Sono previste 2 Prove Intercorso (40% del voto dell'esame orale) + 2 HomeWork (20% del voto dell'esame orale) individuali.

### **Testi di riferimento**

Cloud Computing for Science and Engineering di Ian Foster e Dennis B. Gannon - MIT Press - <https://cloud4scieng.org>

Slide delle lezioni e materiale per approfondimento in piattaforma di e-learning.

## Course objectives

The course is an introduction to programming applications in the cloud computing environment with references to the high performance computing and distributed applications. The course contains an introduction to the main cloud computing architectures as Infrastructure as a Service, Platform as a Service and Software as a Service. Part of the course is devoted to the introduction to the services offered by Amazon Web Services and to the interfacing the cloud with single board computers and Arduino or similar microcontrollers. During the lessons, continuous references are made to the technologies for the development of web applications, mobile applications and mobile multi-platform hybrid applications, the management and representation of spatial data and distributed / wearable pervasive systems. The course contains an introduction to the Python language, used for the development of software projects and hardware in the activities (single or group) in the Laboratory that are an integral part of the course. The course has a typically practical / application setting designed to solve concrete problems and develop software that is actually usable.

**Understanding skills:** The student must demonstrate knowledge and understanding of the fundamentals of cloud technologies and the Internet of things, with particular regard to complex and distributed applications, web service-based application development and analysis methodologies, Python programming language and its application context, methodologies and software development techniques in this context.

**Applicative skills:** The student must demonstrate his / her knowledge acquired to solve computational problems such as cloud applications or web /mobile / embedded hybrid applications based on cloud technologies in concrete contexts, to develop design solutions and analyze them from the point of view of architecture and usability, to consciously use one or more programming languages to implement a complex web application. These capabilities also extricate in a thorough and aware use of the tools offered by cloud computing and the Internet of things and different application frameworks.

**Judgment autonomy:** The student must be able to know how to independently evaluate the results obtained from the application developed in both design and software implementation, evaluating its effectiveness and efficiency.

**Communication skills:** The student must be able to draw up a project submission report and document its implementation, including working in a group, using advanced writing / documentation tools, with particular reference to the use of productivity tools for developers as systems for Shared Publishing (Google Drive), Remote Collaboration (Slack), Production Management (Trello), Online Repository Management (GitHub), Cloud Infrastructure as a Service (Amazon AWS) Using the Web Technology Terminology Correctly Also in English.

**Learning skills:** The student must be able to update and deepen autonomously specific web technology topics and applications, including accessing databases, online document repositories, software and application libraries, and other ways available from the network.

## Contents

- Managing data in the cloud
- Computing in the cloud
- Cloud as platform
- Building your own cloud
- Security and other topics

## Syllabus

- Storage as a service
- Using cloud storage services
- Distributed Databases: CosmosDB
- Computing as a service
- Using and managing virtual machines
- Using and managing containers
- Scaling deployments
- Singularity: a Container System for HPC Apps
- Serverless and Function as a Service
- Data analytics in the cloud
- Streaming data to the cloud
- Machine learning in the cloud
- Supplement: Autoencoders and Manifold Learning
- The Globus research data management platform
- Building your own cloud with Eucalyptus (with Rich Wolski)

Building your own cloud with OpenStack (with Stig Telfer)  
Building your own SaaS  
Security and privacy  
History, critiques, futures

## **Language**

Italian. If an Erasmus student is attending the class, the course will be taught in English.

## **Exam**

The purpose of the verification procedure is to quantify the level of achievement of the training objectives previously indicated.

The verification procedure is indicated precisely in the e-learning platform of the Department of Science and Technology. In summary: around 1/3 of the course, students submit a draft proposal reinforced by a presentation consisting of no more than 3 slides and a duration not exceeding 3 minutes. The proposal may be individual or group consisting of no more than 3 students. The proposal is amended by the teacher, rejected or assigned as an exam project. About 2/3 of the course is organized with a presentation of the projects in which each student or group of students illustrates the state of the art of the project and discusses with teachers and other students about its potentialities and weaknesses.

At the end of the course is organized a final presentation that has no value of examination but a moment of communication of the work done. The project must be released in open source rather than Apache 2.0 on a public repository such as, for example, GitHub at least 2 days before the exam date and must be online at the end of the exam. The verification procedure consists of an individual oral exam (60% of the vote), the presentation of the project part of its competence (20% of the vote), project demonstration (20% of the vote). There are 2 Test Examinations (40% of the oral exam) + 2 HomeWork (20% of the oral exam) individual.

## **Book and other teaching material**

Cloud Computing for Science and Engineering by Ian Foster and Dennis B. Gannon - MIT Press - <https://cloud4scieng.org>

Teaching material and references for individual home works are available on the e-learning platform.