

## Chi è BioDec

Nata nel 2002 come spin-off di un gruppo accademico, con la missione di fornire software e consulenza in **bioinformatica**, dal 2006 l'offerta dei servizi si è ampliata nell'ambito dell'*Information Technology (IT)*, poiché le tecnologie sviluppate e le competenze acquisite si sono rivelate proficue oltre la bioinformatica e applicabili per risolvere molti problemi dell'IT moderno.

La componente di servizi di consulenza IT è successivamente cresciuta negli anni e ad oggi il supporto e lo sviluppo di infrastrutture IT sono tra le attività principali dell'azienda: esse vengono portate avanti utilizzando quell'insieme di tecniche e di pratiche che vanno sotto il nome di **DevOps**.

Fin dalla fondazione, BioDec ha scelto di usare **esclusivamente software libero e standard aperti**.

## I corsi BIODEC per le infrastrutture IT

1. Ansible
2. Logging / Alerting / Monitoring
3. PostgreSQL base
4. PostgreSQL avanzato
5. Python
6. Git
7. Gitlab
8. Linux
9. Secure coding

## Ansible

Automatizzare le attività ripetitive e standardizzare gli ambienti è un problema che investe il mondo IT sia dal lato della gestione dei sistemi (Ops) che dal lato dello sviluppo (Dev). Il proliferare delle piattaforme e in particolare l'uso delle soluzioni *Cloud* ha accentuato il problema, rendendo necessario avere strumenti veloci, affidabili e facili per la gestione di configurazioni e per il continuo rilascio di codice.

Ansible è uno strumento per orchestrare l'installazione e la configurazione di sistemi ed applicazioni: permette di definire lo stato di uno o più server in modo prevedibile, replicabile e consistente. Simile in questo a software come Puppet, Chef, Saltstack o CFEngine, esso basa la propria filosofia su una parola chiave: la semplicità d'uso.

A differenza di altri sistemi, Ansible non necessita di altro che di una connessione SSH tra la macchina che funge da controllore e i server da controllare, e tutto si gestisce con semplici direttive di configurazione contenute in file testuali. Ancora: è una soluzione *agentless* per cui non è necessario installare nessun client sugli *hosts* da configurare, ma solo Python e - cosa che non guasta - è anche un prodotto che presenta una curva di apprendimento molto agevole.

Il corso ha come obiettivo imparare ad usare Ansible per fare partire, fermare e configurare dei server Linux su Cloud Amazon (AWS); si imparerà come si definiscono con Ansible i profili di configurazione di vari server, corrispondenti ad usi differenti (i.e. web server, database server, server di monitoraggio, eccetera). Si imparerà inoltre come svolgere alcune azioni ripetitive su tutti i server dell'infrastruttura, come ad esempio aggiornare dei pacchetti software, fare deploy di nuovi programmi, eccetera. Infine, nella parte finale del corso si affronteranno alcuni temi specifici dell'automazione su AWS, come Cloudformation.

Il corso è volto ad una platea di sistemisti o programmatori che vogliono iniziare ad automatizzare la gestione della propria infrastruttura: il corso è stato svolto ormai in 4-5 edizioni, con un'ottima risposta da parte dei nostri clienti. È un corso sia per principianti che per chi ha provato ad utilizzare Ansible ma non è ancora riuscito a farlo diventare una pratica quotidiana. È possibile avere dei moduli di mezza giornata più specifici su aspetti tecnici come l'integrazione con le tecnologie di AWS, Cloudformation, eccetera.

**È un corso di una giornata.**

## Corso di Monitoraggio, Alerting e Logging

Un corso per apprendere un linguaggio comune su temi sempre più attuali, e per imparare a conoscere nuovi strumenti. Nel prossimo futuro, i programmi e i sistemi che si andranno a realizzare saranno sempre più distribuiti, è quindi necessario conoscere le tecniche migliori per gestirli.

Le moderne infrastrutture IT hanno avuto un'esplosione, che continuerà nei prossimi anni, in termini di complessità e di quantità (di server, di servizi, di tipologie di soluzioni - *public cloud*, *private cloud*, *VPS*, *on premise*, *as-a-service* eccetera). Per gestire al meglio tale complessità è necessario conoscere e utilizzare i nuovi strumenti che sono stati sviluppati in anni recenti dalle comunità del *free software* a proposito della gestione di **sistemi distribuiti**.

Questo corso si propone di essere un'introduzione sia teorica che pratica, agli aspetti di gestione di una moderna infrastruttura IT attraverso tre temi: il **Logging**, il **Monitoring** e l'**Alerting**.

- Il tema del *logging* è legato ad una corretta gestione degli eventi del sistema sotto osservazione, ed è legato ad aspetti quali la conformità a normative, la possibilità di dimostrare il corretto funzionamento in caso di guasti, ed in generale ad aspetti di *compliance*.
- Il tema del *monitoring*, soprattutto applicativo, è ancora poco conosciuto, ed è un peccato perché esistono numerose librerie (nei più diffusi linguaggi di programmazione) e numerosi strumenti coi quali si può instrumentare il codice che si produce. L'strumentazione funge da test continuo del codice, e permette di validarlo nel corso della propria esecuzione.
- Infine l'*alerting* verte su come configurare, gestire e mantenere correttamente il sistema esperto che, sulla base dei dati raccolti con le prime due tecniche (quindi, log e metriche) determina se il sistema sotto osservazione si sta comportando in modo corretto o meno, e in caso negativo propaga le informazioni rilevanti ai diretti interessati.

Al mattino è prevista una presentazione su ognuno dei tre temi, intervallata da un gioco, chiamato ALMA Game (ALMA is A Logging, Monitoring & Alerting Game - presentato ad XP2014) e altre attività di gruppo. Nel pomeriggio, per ciascuno dei tre temi si mostreranno i principali sistemi *free* utilizzabili, accennando alla loro installazione, al caso di uso principale e alle peculiarità del prodotto. Riguardo al logging si accennerà a sistemi come **Graylog2** e allo *stack logstash / elasticsearch / kibana (ELK)*, riguardo al monitoring a sistemi basati su **Prometheus, Influxdb** e **Grafana**, e riguardo all'*alerting* a sistemi classici basati sulla famiglia **Nagios / CheckMK**.

Il corso può essere svolto sia da principianti che da esperti, è un corso intermedio che ha aspetti di interesse anche per le persone più esperte, soprattutto nella sessione del pomeriggio, ove si mostrano gli strumenti, e che permetterà ai principianti di avere una visione di insieme di cosa permettono gli strumenti proposti e soprattutto quando e come adottarli.

## Corso base PostgreSQL

PostgreSQL è un database free di grande caratura tecnica, estremamente concorrenziale rispetto alle alternative *free* e commerciali in quasi tutti gli scenari di uso. Recentemente sta prendendo piede anche all'interno di strutture finanziarie, economiche, mediche e legali, dove gli aspetti di tutela dei dati e delle transazioni è di critica importanza.

Il corso è pensato per una platea di sistemisti / programmatori / capi progetto che pensino di utilizzare PostgreSQL al posto di database commerciali in contesti critici (finanziari, transazionali, grandi moli di dati) e che abbiano competenze anche limitate di gestione di basi di dati. L'obiettivo del corso è dare un quadro chiaro del funzionamento di PostgreSQL e delle sue peculiarità, con informazioni su come fare partire e fermare il sistema, come configurarlo per scenari semplici (non replicati) e come effettuare delle operazioni di diagnostica e di manutenzione.

Sarà mostrato il funzionamento di PostgreSQL su un server Linux Debian Stretch, nella versione ultima disponibile (al momento la versione 11). **È un corso di due giornate** ed è propedeutico al successivo.

### Introduzione a PostgreSQL

- Perché PostgreSQL

### Architettura di PostgreSQL

- Architettura
  - Files
  - Processi
  - Memoria

### Struttura del database PostgreSQL

- Database
- Relazioni
  - Tabelle
  - Indici
  - Viste
- Partizionamento
- Tablespace
- Transazioni
- Schemi

## Configurazione di PostgreSQL

- Configurazione
  - Parametri di autovacuum
  - Parametri dei writers
  - Parametri di checkpointing
  - Parametri di debug
  - Parametri del linguaggio
  - Parametri di logging
  - Parametri relativi alle risorse
  - Parametri delle transazioni
  - Parametri dei dati temporanei
  - Parametri di vacuum
  - Parametri del WAL

## Manutenzione di PostgreSQL

- Manutenzione
  - VACUUM
  - ANALYZE
  - Profiling
- Troubleshooting

## Installazione di PostgreSQL

- Installazione

## Corso avanzato PostgreSQL

Molti applicativi di largo uso adottano PostgreSQL come database (ad esempio, ma non solo, Redmine, Gitlab, eccetera) e in numerosi casi il *database server* è (giustamente) scelto quando serve un sistema transazionale: però spesso ci si accontenta di un *setup* minimale, e non si sfruttano le potenzialità di PostgreSQL per costruire sistemi ridondati e robusti, o vere e proprie architetture di *disaster recovery*.

PostgreSQL è un fantastico progetto *free software*, che mette a disposizione di chiunque un database server ricco di funzionalità, estremamente stabile e ormai rodato anche in contesti *enterprise* dove si richiedono prestazioni elevate e robustezza assoluta. La nostra soluzione è un corso che mostri come funzionano gli aspetti architetturali chiave del sistema (*WAL* e *MVCC*) e come da pochi semplici concetti si possa proseguire nella costruzione di architetture complesse per affrontare gli aspetti di scalabilità, ridondanza e disaster recovery.

È un corso volto ad una platea di sistemisti / programmatori / capi progetto che pensino di utilizzare PostgreSQL al posto di database commerciali in contesti critici (finanziari, transazionali, grandi moli di dati) o persone che usano PostgreSQL (magari a loro insaputa) e che vogliono capire meglio come funziona e adottarlo anche in configurazioni più complesse della modalità single server. L'obiettivo primario è pertanto quello di dare una panoramica quanto più completa delle possibilità di usare PostgreSQL in situazioni reali complesse, ove vi sono dei requisiti di ridondanza o di disponibilità del servizio molto stringenti, o per realizzare una soluzione di disaster recovery.

Si passeranno in rassegna le alternative esistenti, cercando di inquadrare quale utilizzare a seconda degli scenari, quali strategie adottare, quali soluzioni “canoniche” sono disponibili, come configurarle, e come garantirsi del corretto funzionamento del sistema. **È un corso di una giornata.**

### SQL tuning di PostgreSQL

- Perché gli indici
- Come è fatto un B-tree index
- Query plan
- Please EXPLAIN
- Ottimizzazioni

### Backup di PostgreSQL

- Backup
  - Snapshot
  - pg\_dump

- pg\_dumpall
- Formato del dump
- pg\_restore
- Continuous Archiving e PITR
  - pg\_basebackup
  - Ripristino da un archivio continuo
  - Limitazioni

### **Tools di PostgreSQL**

- Connection pooling
  - Pgpool-II
  - pgbouncer
- Benchmark
  - pgbench
- Estensioni

### **Replica di PostgreSQL**

- Replica
  - Tipologia delle soluzioni
  - Soluzioni interne
  - Failover
  - Bilanciamento
  - Parametri di configurazione
- Soluzioni basate su software esterno

## Programmazione Python

È un corso introduttivo su Python pensato per una platea che non conosca il linguaggio, ma che provenga da una formazione scientifica. Parte del materiale infatti si svolge sulle librerie scientifiche di Python.

**È un corso di due giornate.**

## Corso Git

È un corso per iniziare ad usare Git ed è volto a programmatori e sistemisti. Nel corso si mostra come integrarsi con Gitlab e Github. È un corso che è stato già proposto in diverse aziende che hanno delle aree dedicate allo sviluppo interno di software, con una risposta molto buona.

**È un corso di una giornata.**

## Corso Gitlab

È un corso per manutentori e utenti amministratori di Gitlab. Esso verte principalmente sugli aspetti sistemistici relativi alla gestione di un server Gitlab operativo e sulla configurazione della *continuous integration* di Gitlab (toccando quindi i diversi modi, basati su Docker, macchine virtuali, eccetera).

**È un corso di una giornata.**

## Corso Linux base

È un corso per sistemisti che iniziano ad avvicinarsi a Linux, tipicamente proveniendo da ambienti Microsoft. Il tema è molto ampio, e si tratta di come è organizzato il file system, del funzionamento della rete, della gestione dei processi e della memoria e di come fare troubleshooting. A seconda del livello dei discenti, può essere erogato su **una, due o anche tre giornate.**



## Corso di *secure coding*

Il corso offre un'introduzione ai principi di sicurezza informatica per un gruppo di programmatori e sistemisti. I temi trattati vertono, nella prima metà del corso, sui cardini tecnologici della sicurezza informatica, ovvero:

- protocolli
- controllo degli accessi
- crittografia

La seconda parte del corso verte sulle tecniche da utilizzare per il *design* di un sistema sicuro, e quindi saranno introdotti:

- STRIDE,
- *attack trees*,
- *attack libraries* e OWASP.

Il corso si conclude con un *overview* dei punti salienti introdotti dal GDPR, ovvero il regolamento europeo 2016/679 del parlamento europeo e del consiglio, entrato ufficialmente in vigore il 24 maggio 2016, che prevede, fra le altre cose, il principio di *privacy by design*.

Il corso intende avere un'impostazione in parte teorica, nella prima parte, quella più dedicata agli aspetti ingegneristici (intesi come *software engineering*) e in parte pratica, nella seconda parte, dedicata alla progettazione. Il corso fornirà delle tecniche concrete che i partecipanti potranno adottare nella loro pratica quotidiana immediatamente.

**È un corso di una giornata.**

## Contatti

### Internet

- <http://www.biodec.com/>
- [info@biodec.com](mailto:info@biodec.com)

### Fonia

- Telefono (+39)-051-0548263
- Fax (+39)-051-7459582

### Indirizzo

- BioDecS.r.l. Via Calzavecchio 20/2, I-40033, Casalecchio di Reno (BO), Italia.