# Alejandro Bernal Collazos

## Interests

Software Engineering

DevOps

Web Development

Internet Of Things

Continuous Deployment

## Contact Information


| | |
|---|---|
| **Phone** | (+0054 9 351 27 32 9 83) |
| **Email** | alejandro@alejandro.bio |
| **Birth date** | September the 16th of 1986 |
| **Website** | https://www.alejandro.bio |

# Professional Experience

## DevOps June 2019 --> Current time


**Position** **DevOps**

**Company** **IncluIT**

**Client** **Naranja**

**Project** **DevOps Team**

**Activities**

**Infrastructure**

- Work with **Jira** to track the requirements, new features and bugs
- **Design** service and application **infrastructure**
- Manages **pipeline** with **Gitlab** to deploy app/service into production
- Configures the monitor tools with **DataDog**
- Configure the **kubernetes clusters** for these environments
    - Develop
    - Staging
    - Production
- Create bash **scripts** to automate processes like backups, cleaning and notification

**Continuous Integration / Delivery / Deployment with Jenkins**

- Implement **GitLab** to orchestrate these process
    - **Continuous Integration**
    - **Continuous Delivery**
    - **Continuous Deployment**
- Work side by side with development teams in order to standardize and create **pipeline as code** with **Python** for their componentes using **Docker**
    - FrontEnd
    - Backend
    - Database
- Setup, configure and maintain **Docker Registry** to store the images from the Continuous Delivery process
- Setup, configure and maintain **Kubernetes Clusters** for these environments
    - Development
    - Staging
    - Production

**Support**

- Help the development with
    - How to perform deployments into production
    - How to request for development environments
    - Coach and teach about the architecture of the infrastructure
        - Development environments
        - Production environments

## Teacher

**January 2019 --> Current time**

ASYS

**Position** **Entrepreneur**

**Company** **ASYS**

**Client** **Santex - IncluIT**

**Project Activities** **Courses on DevOps**

**Teaching DevOps Culture**

- What DevOps is and where does it comes from
- How to make a frontEnd with HTML, CSS and JS from scratch
- How to make an api with NodeJS from scratch using express
- How to work with mysql databases
- How to setup a linux environment
- How to setup infraStructure with Terraform in AWS
- What is docker and how to create docker images/containers for the api made in NodeJS
- How to setup a docker registry
- What is kubernetes and how to create a cluster with 3 nodes (1 master and 2 slaves)
- What is Jenkins and pipeline as a code
- How to create a pipeline in Jenkins
- How to setup ELK
- How to configure the telemetry for
    - Application
    - Infrastructure
    - Business
- How to use chaos toolkit to inject failures in the system on purpose

## DevOps

**June 2017 --> December 2018**

santex

GRENZEBACH

**Position** **Site Reliability Engineer (DevOps)**

**Company** **Santex**

**Client** **Grenzebach**

**Activities**

**Infrastructure**

- Work with **Jira** to track the requirements, new features and bugs
- **Design** service and application **infrastructure**
- Defines infrastructure within **Ansible** files
- Manages **pipeline** with **Jenkins** to deploy app/service into production
- Configures the monitor tools with Elasticsearch Logstash and Kibana (ELK) for **Monitoring**
- Configure the **kubernetes clusters** for these environments
    - Develop
    - Staging
    - Production
- Create bash **scripts** to automate processes like backups, cleaning and notification

**Continuous Integration / Delivery / Deployment with Jenkins**

- Implement **Jenkins** to orchestrate these process
    - **Continuous Integration**
    - **Continuous Delivery**
    - **Continuous Deployment**
- Work side by side with development teams in order to standardize and create **pipeline as code** with **Groovy** for their componentes using **Docker**
    - FrontEnd
    - Backend
    - Database
- Setup, configure and maintain **Docker Registry** to store the images from the Continuous Delivery process
- Setup, configure and maintain **Kubernetes Clusters** for these environments
    - Development
    - Staging
    - Production

**Support**

- Help the development with

- How to perform deployments into production
- How to request for development environments
- Coach and teach about the architecture of the infrastructure
    - Development environments
    - Production environments

## DevOps February 2017 -> June 2017



**Position** DevOps

**Company** IncluIT

**Client** McAfee

**Activities**

**Infrastructure**

- **Design** service and application infrastructure
- Defines infrastructure within **Terraform** files
- Manages **pipeline** with **TeamCity** to deploy app/service into production
- Configure testing phases in the pipeline
- Configures the monitor tools with Elasticsearch Logstash and Kibana (ELK) for **Monitoring**
- Implements **Continuous Integration, Continuous Delivery and Deployment** in the pipeline
- Configure the **kubernetes clusters** for these environments
    - Develop
    - Staging
    - Production
- Manage **AWS**
    - Manage
        - Elastic Kubernetes Services

- Elastic Container Registry

**Support**

- Support for the development team with the implementation of the pipeline
- **Troubleshoot** development environment issues

## Developer — June 2016 -> February 2017



**Position** **Developer within a DevOps Culture**

**Client** **AppDirect.com**

**Activities**

**Development**

- Create new features made with **Spring**, like new services, and integrations with payment gateways.
- Create new features made with **Wicket** (an apache java Framework to develop web applications) like new windows that handle new processes and perform actions on the server side.
- Make **unit tests** for every part of the code that were modified.
- Make **automated tests** for every new feature with **Selenium**.
- Use **GitHub** and **Jenkins** in order to build the new code and integrate it into the master branch to be delivered into production.
- Use **Maven** to manage package dependencies, run the legacy application locally.
- Use **Docker** to run the legacy application within a container, to avoid modification in the local machine.

- Use **AWS machine instances EC2** in order to perform automated tests before delivering a new feature into production.
- Use **JPA** in order to persist data against the DB.
- Perform code review for modifications made by my colleges.

**Bug Fix**

- Troubleshoot and solve issues with the legacy application (monolith), based on **Wicket**, **Spring**, **JavaScript** using **MySQL** database.
- Make unit tests for any change in order to cover such changes.
- Make automated tests for any new change in the existent application that impact the user interface such as the business logic

## DevOps

**February 2014 -> June 2016**



**Position** **Developer and later System Administrator within a DevOps culture**

**Client** **LATAM.com**

**Activities**

**Development**

- Support and Maintain the legacy website based on **Perl**
- Create backend services based on **Java** with the Spring Boot framework
- Create and setup **Docker** for the backend services
- Work with **Git** to perform version control and **Git Flow** as the branching model

- Work with **Jira** for work tracking
- Work with **Scrum** as the Agile practice to create new feature within the development team
- Work with **Artifactory** to manage legacy backend services based on Spring
- Work with **Splunk** for **Monitoring** to perform log aggregation and notifications

**Automation - Configuration / Setup and Maintenance**

- **Scripting** Create and maintain scripts (made in **Bash** and **Python** mainly) to make Deployments (of **java** artifacts from archives into the **tomcat** servers of a given machine)
- Use **SVN** and **GIT** to version control **Chef** cookbooks
- Request machines to the cloud provider **Softlayer** through a self service layer that developers can use by their own
- Create, configure and maintain **Linux** user accounts (**CentOS** for development and **RedHat** for production)
- Create, configure and maintain **Linux** machines (CentOS for development and RedHat for production environments)
- Install and Configure servers (**Apache** http, dns, ldap ,Haproxy, Tomcat)
- Configure Linux **Firewalls**
- Configure Linux **Tunnels**
- Setup **Cron** rules to execute jobs on given schedules and automatically
- Configure Apache HTTP server
    - Setup reverse proxy rules
    - Setup virtual hosts
- Create, Maintain and configure **MySQL DB cluster**
- Configure **Haproxy** server
    - In order to perform load Balance
    - To promote service discovery
- Configure **Chef**
    - Create and Maintain Cookbooks
- Configure **Rundeck Server**
    - Define Jobs to be executed on given machines, like restart a given set of machines defined by a role
- Setup and maintain **Nagios Server** for Infrastructure **Monitoring**
    - To perform monitoring in the development environments and production environments
- Version Control Server **SVN**
    - Administration

**Bug Fix and diagnosis**

- Troubleshoot and solve issues on
    - Infrastructure
    - Application
    - Deployment
    - Java Web Services
    - JavaScript applications

**Support**

- Help the development teams in this tasks
    - How to perform deployments into production
    - How to request for development environments
    - Coach and teach about the architecture of the infrastructure
        - Development environments
        - Production environments

**User Accounts**

- Configure LDAP servers in order to
    - Create - Modify and Delete
    - Groups
    - Users
    - Domains
    - Attributes

## Developer

**February 2012 -> February 2014**

**Position** **Developer**

**Client** **AA.com American Airlines**

**Activities**

**Development**

- **Java** create web services based on spring that could interact via SOAP with clients and providers of information
- **Logging** we used Log4j in order to perform the logs of our web services
- **Unit Testing** with JUnit for the unit tests
    - Test for functions in the code
- **SonarQube** usage for static code analysis
    - Coverage of 80% or greater
    - Duplicated code
    - Syntax checks
- **Branching Model** we used a flavor of gitflow based on feature branches in order to implement the continuous integration process
- **Continuous Integration** using the branching model we established a set of steps to guarantee that the code (whether on feature branches, pull requests or new merges into the develop branch) works according to our functional policy
    - Code is able to build
    - All unit tests passed
    - The amount of coverage is within the established parameters (>= 80%)
    - The static code analysis passed properly
        - Check for code duplication
        - Cyclomatic complexity
        - Syntax check
- **Systems Development Life Cycle** The requirements for new improvements or bug fixes came from tickets and according to their priority, some of them were done as hotfixes or some of them worked out using **Agile/Scrum** within a sprint of 2 weeks period and later delivered to **Artifactory** --- using the **semantic versioning** --- as packages,, for a provider (Verizon) in charge of the deployment in a production environment
- **Software Configuration Management** Although we did not have full control over the management of the services in production we had
    - A tool to check which specific version is installed in production and in which server

    - Within the release process we documented on the release notes which changes or tickets were included in a particular version in order to keep track of them
- **Regression Testing** with SOAP UI against the services layer, in order to avoid new changes to break existing functionality

**Support**

- **Maintain** the web services already created as well as the new ones, this implies update the wsdl or service contract in order to match the new feature in the service layer
- **Talk and coordinate** with Development teams based in Dallas issues or usage of our web services
- **Talk and coordinate** with QA team based in India the testing phases on each one of the services

# Formal Education

## National Technical University **January 2008 – January 2009**

City Capital Federal, Buenos Aires. Argentina

Title "Systems Engineering"

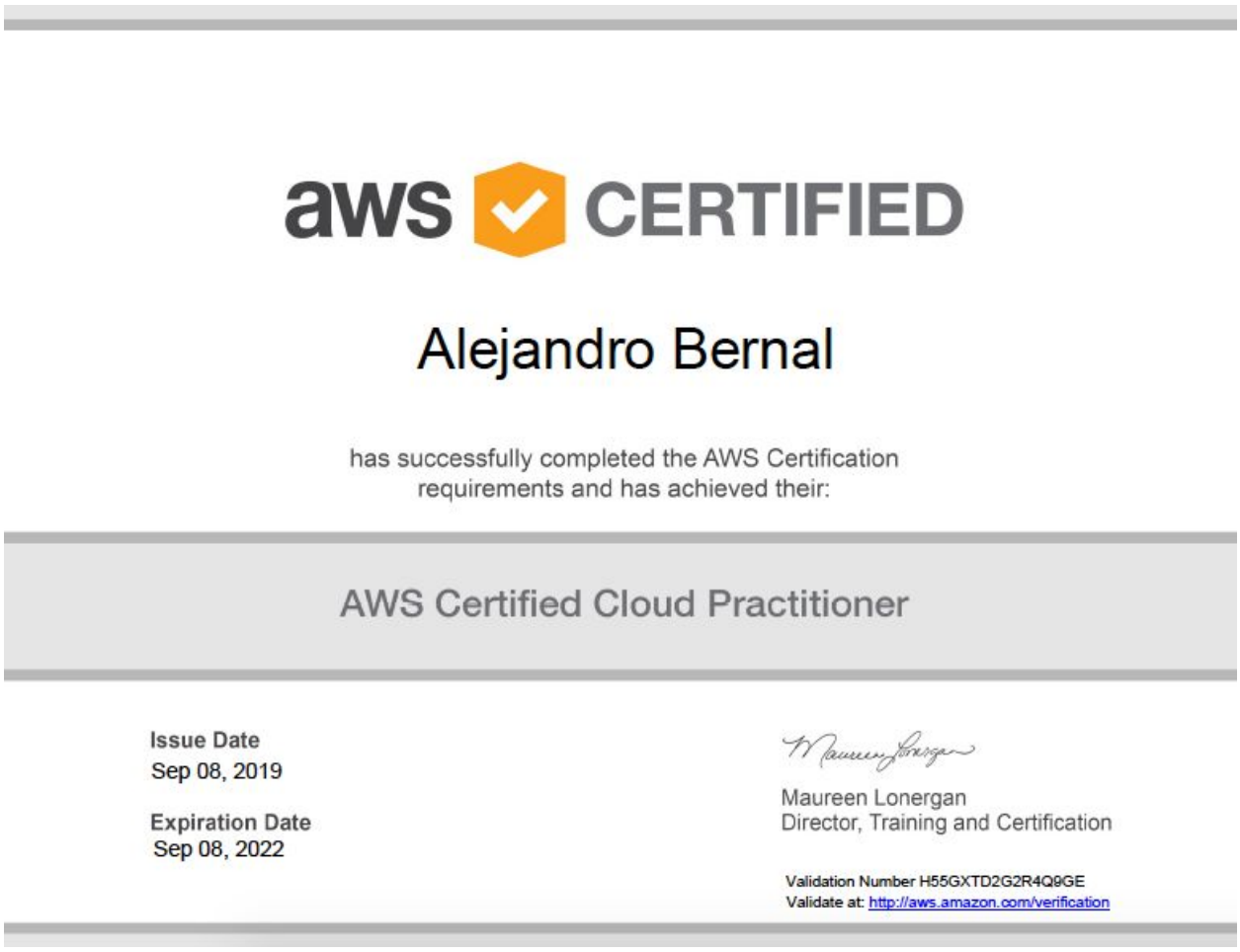
Status **Not Finished**

# Certification

## Oracle Certified Professional, Java SE 6 Programmer — January 2012


## Linux Foundation Certified SysAdmin — May 2017

**AWS Certified Cloud Practitioner** **September 2019**



aws CERTIFIED

Alejandro Bernal

has successfully completed the AWS Certification requirements and has achieved their:

AWS Certified Cloud Practitioner

**Issue Date**
Sep 08, 2019

**Expiration Date**
Sep 08, 2022

Maureen Lonergan
Director, Training and Certification

Validation Number H55GXTD2G2R4Q9GE
Validate at: http://aws.amazon.com/verification

# Languages

## English

Advanced Writing and Speaking

## Portuguese

| | | |
|---|---|---|
| Intermediate | | Writing and Speaking |

### Spanish

| | | |
|---|---|---|
| Mother Language | Advanced | Writing and Speaking |

## Technical Skills

### Programming and Scripting languages

- Java
- SQL
- Php
- Bash
- Perl
- Python
- JavaScript

### Operative Systems

- Linux
  - Ubuntu
  - Fedora
  - CentOS
  - Debian
- Windows
- Mac OS

### Content Management Systems

- Drupal

### Databases

- MySQL

- PostgreSQL

### Web Servers and Web Application Servers

- Apache
- Tomcat
- Nginx

### Configuration Management Tools

- Ansible
- Chef

### Virtualization Technologies

- Docker

### Continuous Integration Servers

- Jenkins
- TeamCity

### Containers Orchestrators

- Kubernetes

### Cloud Services for Infrastructure as a Service IaaS

- AWS
    - Route53
    - API Gateway
    - Elastic Kubernetes Service
    - EC2
    - Elastic Load Balancer
    - Auto Scaling Groups
    - Network Load Balancers
    - Cloud Formations

# Courses

| | |
|---|---|
| Portuguese | [CEPE : Center] March 2013 - January 2015 |
| JAVA Enterprise Edition 5 | [UTN Córdoba] July 2012 - January 2013 |
| UML y UP : Design and Analysis | [Educacion IT] September 2011 - January 2011 |
| Java Hibernate | [Educacion IT] September 2011 - January 2011 |
| Java Web | [Educacion IT] September 2011 - January 2011 |
| Java Programming | [Educacion IT] September 2011 - January 2011 |
| PHP Programmer | [Educacion IT] September 2011 - January 2011 |
| Linux System Administrator | [Educacion IT] September 2011 - January 2011 |