**Position Description**

**Title: DevOps Engineer**

**Job Position Duties and Responsibilities**

**Description:**

A brief outline of the key job functions are :

* As a technical associate primary responsibility would be to abide by the assignments given by the Tech Theme Solutions Inc. reporting manager and primarily engage in ***working on designing and implementing Continue Integration and Continues deployment for many java and .net based applications helping project teams to develop, deploy, test and deliver software packages fast and reliably and working on bash and python script for automation process of release branches of all components at end client “CISCO” in San Jose, CA.***
* Will also be responsible for sending weekly status report and time sheets on the work done to Tech Theme Solutions Inc. Project manager.
* Assignment with roles and project will separately discuss. Primary job duties include design, development, testing, implementation, maintenance and supporting web based and client server-based software applications.

**Job Duties for the position:**

1. **Collaborate and develop a application with QA Team** to know their regression jobs and with the developer to **develop a code for sanity testing** (**20%)**

* Provide **direct analysis** and recommendations to develop code for microservices and regression testing failures for release of our microservices
* These meetings are required to understand the existing processes, create a working knowledge of the existing applications in place, understand the roadmap for the new developments required as per the requirements outlined by the developers and create an effective and efficient solution for the microservice.
* **Technical meetings for Application Development** also discuss the latest technology available to support the requirements and adaptation of the new technology within the company.
* These meetings help us to understand the main areas required by the Business users and the primary goal of using these applications and help in development of Demo of applications and design for high level architectural implementation.
* These meetings also help to develop **the Architecture of our Cisco Application policy infrastructure Controller** for outlining software application process.
* **These meetings help to create a list of User Stories in Rally which should be worked on in the next sprint**.
* Analysis are performed at all levels of total system product to include: concept, design, deployment, test, operation and Release.

1. Work as the **Development Operations Engineer as a Subject Matter Expert** for various new incoming tools and be the point of contact for any cloud or build related solutions when brining in new applications **(20%)**

* ensure that the tools is increasing to help on specific tasks for code testing to increase and integration to Jenkins build that has to work properly and trying out with our code and make sure it has to offer highest quality effects to adopt it.
* Developing the cookbooks on nodes and running CHEF client on them in a Chef environment and Chef recipes to automate infrastructure as a code of **AI**.
* Make decisions on how to store the code and the process to reuse the code by using make file compilation and used for design the application of our project.
* Develop code make file compilation by using GCC compiler for applications and **integrate** properly and ensure that the build is continuous without any issues and failures.

1. Work with **Amazon Web Services** when onboarding a new application in the environments and make sure that the code is being secured when deploying and the right terminology is being used in the contract **(10%)**

* Written Ansible Play books to provision several pre-pod environments consisting of deployment automation, AWS EC2 instance mirroring, web logic domain creations and several proprietary middleware installations.
* Design and Develop **cloud formation as code** is planned before the software project deployment.
* Develop **End to End Support model** for implementing the project for Software development.
* Develop **Release Management** process for application Development and ensure that the right tools are being used to develop and deploy the application upgrades.
* Ensure that the contract includes the **Service Level Agreements** between the vendor and Cisco and act as the first point of contact for the vendor when an issue is raised.

1. Work with **Microservice teams introducing additional new tools** and ensure that is being implemented **(15%)**

* Work with project teams to implement and support the vendor applications.
* Using the **support documentation checklist in WIKI**- ensure that the Cisco policies are being followed and the right documentation is handed to the support team before handoff.
* Work with Project teams introducing additional new microservices to the developer Unit and make sure that the **Regression Management tasks** are planned.
* Implementing and maintaining cloud management solutions including initial and ongoing configuration of related automation, notifications, and reporting capabilities.
* Identify and troubleshoot cloud service events and issues as well as work with cloud service providers to efficiently solve issues or implement workarounds
* Perform technical planning, system integration, verification and validation, evaluate alternatives including cost and risk, supportability and analyze for total systems
* Deployment plan lists all the activities required for implementing the code to Production Cloud environment after development.
* Deployment checklist lists the total number of components moved to production.

1. Be the **Primary Customer Contact for Vbranch, managed device and Cutd** applications and the primary Operational Contact for application teams **(15%)**

* **Validate Software code for the software application in Production**
* Post production support to validate code from the software application
* **Acknowledgement of Remedy or Issue tickets** during the support phase.
* Work with a variety of legacy java based applications and platforms and work with microservice teams to implement or migrate associated components to cloud services
* Perform various **Compliance Activities** on the applications being Supported by Cloud Vendors on an every Sprint Basis.

1. **Evaluate the current Governance processe**s new Automation tools available in the market and create Proof of concept **(5%)**

* **Proof of concept** designs and development for help in software automation tool selection by comparing the performance with latest Ansible automation tool.

1. **Create Technical documentations**, technical design documents to support software application development and **support** the applications **(15%)**

* Technical documentations are the outputs of the design phase. All the **design solutions** are documented for future reference and knowledge transfer sessions.
* Be a part of the **Cloud Implementation Teams** and support any issues with software applications and software debugging when required
* Maintain ( to ensure software applications run optimally as per the best standards) the Linux Instances and the Windows Server Instances on **Amazon webservices Cloud** and ensure they are available for the software applications to run
* Develop HLD (**High level Design**) to capture the overall system design, including the system architecture and database design, DLD (Detail level Design) document has the detailed explanation of what is defined in HLD. DLD explains the top-down design approach for the process including the logic, SQL queries, sequencing of the process.
* Creation of Support document with technical details of software development to transition the changes done during the development to application support for future maintenance.
* Create **System integration WIKI documents** to share with other teams.
* Create a support model document which consists a more detailed description of the applications and microservices deployments, testing, release repositories, scripts used in release in WIKI

**Job Itinerary (Breakdown of major Activities):**

**Collaborate and develop a application with QA Team**: 20%

Design and **Development Operations Engineer as a** Process flows as Subject Matter Expert: 20%

Implement the Cloud Adoption Model: 10%

Ensure Develop and work on new tools with teams to get Optimal Results : 15%

Evaluate current Procedures/Automation Tools/Develop and co-ordinate with Microservice and QA Teams: 20%

Create Technical Software Documentation: 15%

**Specific skillset for performing the job on a day to day basis**

* Cloud Platforms and specially Amazon Web Services or Openstack.
* Must have good understanding and work with:
  + Information Technology Infrastructure Library implemented projects.
  + Building and Executing Automated tests for Web Services
  + Deploying the Java based applications
  + Work on creating regression pipelines for our products.
  + Implementing and using Cloud Technologies for deploying microservices after testing.
  + Knowledge of Agile methodology and waterfall models.
* Utilize:
  + Various Cloud Technologies such as Amazon Webb Services, Openstack etc.
  + Agile Development Process and TDD and Continuous Integration and Deployment.
  + Working knowledge in Rally and JIRA.
  + Linux systems, database administration, automation tools, Devops tools such as GIT, JIRA, Rally, JENKINS, Ant, Maven, Chef, Ansible, Docker, Kubernetes, Code deploy, Cloud Formation etc.;
* On a daily basis, be:
  + Creative and problem-solving mindset;
  + Willingness and ability to work in a fast-paced environment with minimal supervision;
  + Entrepreneurial spirit, sense of personal responsibility, high level of self-motivation and maturity;
  + Prioritization and project-management techniques;
  + Review code and test cases;
  + Sustain and support new test frameworks to support new features, releases, and better utilization;
  + Support and work new code build tools like maven and Ant.
  + Design and demonstrate customer specific use-cases;
  + Create and build documentation and training concepts in wiki;
  + Effective internal and external communication and presentation skills.

**Job Requirements (Educational/Experience)**

Bachelor’s degree in a Computers related major such as Electronics and Communication Engineering or closely related fields or equivalent with three or more years of experience in DevOps Methodologies such as Jenkins, Git, Docker, Ant, Maven, kubernetes and Amazon Web Services.

**OR**

Master’s Degree in Electrical and Electronics

**Coursework Descriptions**

1. **Engineering Mathematics**: Computer science requires many mathematical concepts. Everything comes down to binary, the binary requires conversion to other bases. This is the lowest level in which they are connected. This coursework teaches all the principles behind applications of mathematical and statistical analysis excessively used in the field of computer science and other engineering fields.
2. **Computer Programming:** The core computer programming coursework start with introductory classes, such as operating systems and software engineering.  Computer programmers will develop the basic design of Software and will turn them into reality by writing codes. This coding will make the applications or operating system or any piece of logic to function as desired.  This course focuses on design of good program, Algorithms, Flowcharts, Requirements elicitation, File processing, Database architecture, usage of relational and non-relational databases to generate SQL queries and different programming techniques.
3. **Computer architecture and organization:** This course includes basic operation of computing hardware, how it works, and how it interfaces to software. This course gives a high-level understanding of the role played by compilers, assemblers, instruction sets, and hardware. it also teaches system-level programming and application of computer architectures to programming for performance.
4. **Data Structures:** Data structures provide a means to manage large amounts of data efficiently for uses such as large [databases](https://en.wikipedia.org/wiki/Database) and [internet indexing services](https://en.wikipedia.org/wiki/Web_indexing). Usually, efficient data structures are key to designing efficient [algorithms](https://en.wikipedia.org/wiki/Algorithm). Some formal design methods and [programming languages](https://en.wikipedia.org/wiki/Programming_language) emphasize data structures, rather than algorithms, as the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both [main memory](https://en.wikipedia.org/wiki/Main_memory) and [secondary memory](https://en.wikipedia.org/wiki/Secondary_memory). Data structures are generally based on the ability of a computer to fetch and store data at any place in its memory, specified by a [pointer](https://en.wikipedia.org/wiki/Pointer_(computer_programming)) a bit string, representing a [memory address](https://en.wikipedia.org/wiki/Memory_address), that can be itself stored in memory and manipulated by the program. Thus, the [array](https://en.wikipedia.org/wiki/Array_data_structure) and [record](https://en.wikipedia.org/wiki/Record_(computer_science)) data structures are based on computing the addresses of data items with [arithmetic operations](https://en.wikipedia.org/wiki/Arithmetic_operations); while the [linked data structures](https://en.wikipedia.org/wiki/Linked_data_structure) are based on storing addresses of data items within the structure itself. Many data structures use both principles, sometimes combined in non-trivial ways.
5. **Computer Networks:** This course covers the following topics. The structure and components of computer networks, packet switching, layered architectures, TCP/IP, physical layer, error control, window flow control, local area networks (Ethernet, Token Ring; FDDI), network layer, congestion control, quality of service, multicast.
6. **Computer Programming Lab**: Computer Programming Lab course introduces computer hardware, operating systems and commonly used software packages. This program also explains problem solving and program construction using top-down design, data abstraction, and object-oriented methodologies. Focus is given on creating program module using a programming language or scripting language and using a database or file to store and process the data. Application of performance tuning techniques to generate most efficient module is focused throughout the program.
7. **Microprocessors and Microcontrollers:** This course explains the basic architecture of microprocessors and microcontrollers, their interactions to peripheral devices and use of microprocessors and controllers in the design of these systems. Topics include Semiconductor memory devices and systems, microcomputer architecture, assembly language programming, I/O programming, I/O interface design, I/O peripheral devices and data communications.
8. **Computer Graphics through C++:** Computer graphics are [pictures](https://en.wikipedia.org/wiki/Pictures) and [films](https://en.wikipedia.org/wiki/Film) created using [computers](https://en.wikipedia.org/wiki/Computer). Usually, the term refers to computer-generated [image](https://en.wikipedia.org/wiki/Image) data created with help from specialized graphical hardware and [software](https://en.wikipedia.org/wiki/List_of_3D_computer_graphics_software). It is a vast and recent area in [computer science](https://en.wikipedia.org/wiki/Computer_science). Some topics in computer graphics include [user interface design](https://en.wikipedia.org/wiki/User_interface_design), [sprite graphics](https://en.wikipedia.org/wiki/Sprite_(graphics)), [vector graphics](https://en.wikipedia.org/wiki/Vector_graphics), [3D modeling](https://en.wikipedia.org/wiki/3D_modeling), [GPU](https://en.wikipedia.org/wiki/GPU) design, [implicit surface](https://en.wikipedia.org/wiki/Implicit_surface) visualization with [ray tracing](https://en.wikipedia.org/wiki/Ray_tracing_(graphics)), and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), among others. Computer graphics is responsible for displaying art and image data effectively and meaningfully to the user. Programming Languages used for this are C++, C, etc.
9. **Database Management Systems:** A database management system (DBMS) is a [computer program](http://psychology.wikia.com/wiki/Computer_program) (or more typically, a suite of them) designed to manage a [database](http://psychology.wikia.com/wiki/Database) (a large set of structured [data](http://psychology.wikia.com/wiki/Data)), and run operations on the data requested by numerous clients. Typical examples of DBMS use include [accounting](http://psychology.wikia.com/wiki/Accounting), [human resources](http://psychology.wikia.com/wiki/Human_resources) and customer support systems. Originally found only in large organizations with the [computer](http://psychology.wikia.com/wiki/Computer) hardware needed to support large data sets. There are different types of DBMS products: relational, network and hierarchical. The most widely commonly used type of DBMS today is the Relational Database Management Systems (RDBMS).
10. **Operating Systems:** An operating system (OS) is [system software](https://en.wikipedia.org/wiki/System_software) that manages [computer hardware](https://en.wikipedia.org/wiki/Computer_hardware) and [software](https://en.wikipedia.org/wiki/Computer_software) resources and provides common [services](https://en.wikipedia.org/wiki/Daemon_(computing)) for [computer programs](https://en.wikipedia.org/wiki/Computer_program). All computer programs, excluding [firmware](https://en.wikipedia.org/wiki/Firmware), require an operating system to function. [Time-sharing](https://en.wikipedia.org/wiki/Time-sharing) operating systems [schedule tasks](https://en.wikipedia.org/wiki/Scheduler_(computing)) for efficient use of the system and may also include accounting software for cost allocation of [processor time](https://en.wikipedia.org/wiki/Scheduling_(computing)), [mass storage](https://en.wikipedia.org/wiki/Mass_storage), [printing](https://en.wikipedia.org/wiki/Printer_(computing)), and other resources. For hardware functions such as [input and output](https://en.wikipedia.org/wiki/Input_and_output) and [memory allocation](https://en.wikipedia.org/wiki/Memory_allocation), the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and frequently makes [system calls](https://en.wikipedia.org/wiki/System_call) to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer from [cellular phones](https://en.wikipedia.org/wiki/Cellular_phone) and [video game consoles](https://en.wikipedia.org/wiki/Video_game_console) to [web servers](https://en.wikipedia.org/wiki/Web_server) and [supercomputers](https://en.wikipedia.org/wiki/Supercomputer). Types of Operating Systems Batch operating system, Time-sharing operating systems, Distributed operating System, Network operating System, Real Time operating System.
11. **Parallel and Distributed Systems:** Distributed systems are groups of networked computers, which have the same goal for their work. The terms "[concurrent computing](https://en.wikipedia.org/wiki/Concurrent_computing)", "[parallel computing](https://en.wikipedia.org/wiki/Parallel_computing)", and "distributed computing" work. The terms "[concurrent computing](https://en.wikipedia.org/wiki/Concurrent_computing)", "[parallel computing](https://en.wikipedia.org/wiki/Parallel_computing)", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

* In parallel computing, all processors may have access to a [shared memory](https://en.wikipedia.org/wiki/Shared_memory_architecture) to exchange information between processors.
* In distributed computing, each processor has its own private memory ([distributed memory](https://en.wikipedia.org/wiki/Distributed_memory)). Information is exchanged by passing messages between the processors.

1. **Software Engineering:** Software engineering is a process of designing, constructing, and testing software by analyzing needs of end user. In software engineering, programs are written and designed for computers or other electronics products. It ensures that the application is developed correctly, consistently, on time and in budget within the given requirements. Objectives of Software Engineering are to provide insights about factors affecting the market growth, to analyze the Software Engineering Market based porter’s five force analysis etc., to provide country level analysis of the market with respect to the current market size and future prospective. An advanced course in the systematic approach to the specification, development, operation, maintenance, and retirement of software. Topics include formal specification tools, developmental strategies, software metrics, verification and validation techniques.
2. **Team Management**: Team management comprises of several activities, which contains planning of project, deciding scope of software product, estimation of cost in various terms, scheduling of tasks and events, and resource management. An advanced course in the planning and management of all phases of the computer information systems project, including the creation, execution, and monitoring of system project plans. Project management activities may include:
   1. Project Planning: Software project planning is task, which is performed before the production of software actually starts.
   2. Scope Management: It defines the scope of project; this includes all the activities, process need to be done in order to make a deliverable software product.
   3. Project Estimation: Project estimation may involve Software size estimation, Effort estimation, Time estimation, Cost estimation
3. **Engineering Economics and Management**: This course helps an individual to understand functional areas of management and duties an individual should perform in an organization and along with the roles and responsibilities of all the key stakeholders within a project.
4. **Data Resource Management**: Data resource management involves the management of files and computer data for businesses and companies. This course helps an individual to learn more about the SQL and Oracle Databases and how to manage the databases within an enterprise.
5. **Communications Networks:** In-depth treatment of planning, designing and managing communications networks; includes feasibility analysis, forecasting, optimizing using network management software. This course enables one to get a higher view of how enterprise networking solutions work and how to implement them.
6. **Management of Information Systems:** The principles and practice of managing corporate information systems, including acquisition, budgeting, development, and personnel issues. This course enables an individual to learn more on the security level for an organization, the types of threats, how to identify such issues and how to resolve security threats and other issues.
7. **Data Communication:** A comprehensive introduction to the field of digital data communications. It provides an understanding of terminology, topology, protocol, and available systems network architecture.

**Benefits**

* Three weeks paid vacation
* Healthcare and Dental Insurance

**Job Type:** Full time employee on W-2

**Job Requirements (Educational/Experience)**

Bachelor’s degree in a Computers related major such as Electronics and Communication Engineering with three or more years of experience in DevOps Methodologies such as Jenkins, Git, Docker, Ant, Maven, kubernetes and Amazon Web Services.

**OR**

Master’s Degree in Electrical Electronics.

**WAGE LEVEL : 2** (**one for education and one for experience/skills as per the prevailing wage determination guidance worksheet**)