

EDUCACIÓN

Trabajo Integrador de Cloud DevOps

INDICE.

1. Objetivos	3
2. Descripción de las tareas.	4
Sprint 1:	4
Sprint 2:	5
Sprint 3:	6
Sprint 4:	7
Sprint 5:	8
Sprint 6:	8
Sprint 7:	9
Sprint 8:	10
Sprint 9:	10
Sprint 10:	11

1. Objetivos.

En este trabajo integrador el propósito será:

Comenzar con la creación o modificación de una aplicación web simple que el docente proveerá escrita en JavaScript, HTML 5 y CSS, esta aplicación no tiene un propósito en sí, salvo utilizarla para cada una de las fases del proyecto, por lo tanto, se les enviara una aplicación web de ejemplo que utilizaran a lo largo de todo el curso.



Una serie de tareas y objetivos a cumplir, siguiendo las metodologías Ágiles y la metodología Scrum. Aquel estudiante que sea asignado como **Product Owner**, será el encargado junto con los miembros de su equipo, de llevar adelante la siguiente serie de tareas en los tiempos que dure la cursada:



2. Descripción de las tareas.

Sprint 1:

- ❖ Creación de Paneles en Trello integrado con Jira y Atlassian.
- ❖ El PO se encargará de determinar las tareas del Dev-Team, pero en conjunto con ellos, no se las impone, sino que se dialogan.
- ❖ Cada uno de los miembros del equipo deberán:
 - Formar parte de un repositorio en GitHub, en este repositorio, el PO se encargará de crearlo e invitar a los miembros del equipo de Desarrollo y al Scrum master.



- Una vez que todos formen parte del proyecto en GitHub, es necesario que el PO cree dentro del repositorio un archivo readme.md especificando los miembros del proyecto, nombre completo y correos electrónicos, sitios de LinkedIn, rol laboral y carrera que cursan en EducaciónIT.
- Cuando todos estos pasos se encuentren listos, es necesario reportar la finalización de la tarea 1 en Trello, el profesor analizará si se ha realizado correctamente.



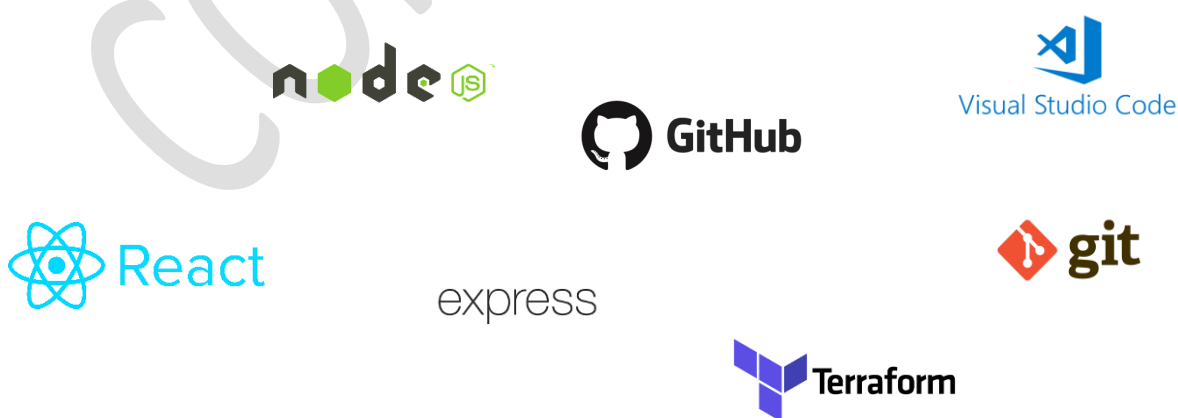
Sprint 2:

- ❖ Una vez creado el repositorio grupal y que estén invitados todos los miembros, será necesario que cada uno de los miembros instale:
 - Git, en su última versión, y crear un directorio para el repositorio de GitHub para luego sincronizarlo.
 - Visual Studio en su última versión, luego, instalen el Plugin de Terraform, el Plugin de HTML Marks, y el Plugin de autocompletado, así como los Plugins de JavaScript, de Node y React. (Los mismos se verán en clase).
 - También es necesario sincronizar Visual Studio Code con la cuenta de GitHub, en este caso, será necesario que la sincronicen y que cada miembro cargue que ha podido sincronizar su Visual Studio con el repositorio de GitHub del proyecto.
 - Luego, será necesario que todos los miembros del equipo instalen NodeJS en su última versión y puedan utilizarlo en Visual Studio.
 - Luego será necesario que instalen el módulo de ReactJS y de Express (Se vera en clase como realizarlo).

(Observación: Si ustedes conocen el procedimiento también pueden realizarlo y así agilizar la entrega.)

Para comprobar que la tarea 2 y cada uno de sus puntos fue realizado correctamente, es necesario que actualicen diariamente en el repositorio de GitHub un archivo de texto con el progreso de cada tarea y minutos de las reuniones de equipo.

Una vez resuelto este aspecto, será necesario actualizar también los estados de las tareas en Trello. El objetivo será, actualizar todo el progreso de manera proactiva en los tableros de Trello y en el Proyecto de GitHub.



Sprint 3:

En esta instancia del proyecto, se les proveerá una aplicación web simple para que la utilicemos como proyecto de deployment a lo largo de las clases, sin embargo, la utilizaremos en breve.

❖ Las 2 primeras actividades son:

- El PO deberá crear una cuenta gratuita en Amazon Web Services, Google Cloud Platform y Microsoft Azure. (Si no dispone de los recursos para crear la misma otro miembro del equipo puede asistir en este proceso)
- Una vez que se haya creado cada una de las cuentas, recuerden actualizar la tarea con capturas del nombre de usuario y la cuenta en Trello.

❖ Una vez que se hayan creado las cuentas en cada uno de los proveedores, será necesario crear, en cada una de las cuentas de AWS, Azure y GCP lo siguiente que se detalla:

- 1 instancia con Windows Server 2019 en Amazon Web Services, con RDP, un Security group abierto a internet, en la zona de disponibilidad A de South América, el nombre debe ser “InstanciaCloudDevopsTeam” y el nombre del team , esta instancia debe estar creada con los tags (año, sistema operativo, nombre del equipo, carrera, Educacionit como institución, proyecto (carrera cloudevops) , también el tag de “Entorno “ con el valor “Development”, debe tener un disco EBS de 30 GB, para confirmar que esta instancia fue creada, debe actualizar con capturas de la misma y de la dirección IP en Trello.
- 1 instancia con Windows Server2012 en Amazon Web Services, con RDP, un Security group abierto a internet, en la zona de disponibilidad B de South América, el nombre debe ser “InstanciaCloudDevopsTeam” y el nombre del team , esta instancia debe estar creada con los tags (año, sistema operativo, nombre del equipo, carrera, EducaciónIT como institución, proyecto (carrera cloudevops) , también el tag de “Entorno “ con el valor “Stage”, debe tener un disco EBS de 30 GB, para confirmar que esta instancia fue creada, debe actualizar con capturas de la misma y de la dirección IP en Trello.
- 1 instancia con Windows Server 2016 en Amazon Web Services con EC2, del tipo t2.micro, el único disponible en la capa gratuita, con acceso RDP, un Security group abierto a internet, en la zona de disponibilidad C de South América, . el nombre debe ser “InstanciaCloudDevopsTeam” y el nombre del team, esta instancia debe estar creada con los tags (año, sistema operativo, nombre del equipo, carrera, EducaciónIT como institución, proyecto (carrera cloudevops), también el tag de “Entorno” con el valor “Producción”, debe tener un disco EBS de 30 GB, para confirmar que esta instancia fue creada, debe actualizar con capturas de la misma y de la dirección IP en Trello

- Por último, será necesario que el DevTeam cree una AMI, Amazon Machine Image de la instancia de Windows Server 2019, y cambien el fondo de pantalla a un fondo que demuestre que el equipo lo creó, pueden utilizar logotipos o diseñar su propio wallpaper, una vez creada esta AMI, que debe tener instaladas las siguientes herramientas en Windows 2019 Server:

- ❖ Docker en su última versión
- ❖ WAMP
- ❖ Kubernetes
- ❖ Visual Studio
- ❖ El ejecutable de Terraform
- ❖ Jenkins en su última versión
- ❖ GIT

Esta AMI, será la prueba de que ustedes pueden crear un entorno de DevOps que luego podrán distribuir a otros miembros de su equipo para que creen instancias en base a esta AMI y así tener una infraestructura de desarrollo alineada.

Una vez realizadas todas estas tareas, y actualizado en Trello, así como en el repositorio de GitHub, podemos dar por concluida cada una de estas tareas.

Sprint 4:

Ahora es el momento de trabajar creando los entornos en GCP con Compute Engine, para ello, lo que realizaremos será:

- crear una Instancia del tipo F1-micro, de primera generación de Ubuntu con contenedores en Google Cloud Platform, con un disco de 100 GB. Seleccionaremos que sea accesible desde cualquier dirección IP del mundo, el nombre de la VM debe ser "InstanciaGCPCloudDevops" y debe tener los tags "Tipo: Contenedor", "Sistema Operativo: Ubuntu" "Equipo: Team x", "Carrera: CloudDevops" "Institución: EducaciónIT".

En esta instancia con contenedores luego probaremos los laboratorios de Docker y las imágenes, por lo tanto, también será un terreno de pruebas para reflejar en nuestro repositorio de github.

En Trello, deberán cargar capturas de que la instancia fue creada con los parámetros especificados y también, que la instancia tiene Docker instalado. En clase, aprenderemos como comprobar que Docker está instalado en nuestra Instancia o VM.

Sprint 5:

Ahora es el momento de trabajar con Microsoft Azure, para ello, utilizaremos:

- Una máquina virtual en Virtual Machines de Microsoft Azure, para esta instancia, crearemos una VM del tipo BSDE, con un Windows 2019 Datacenter, descargaremos y dejaremos en una carpeta en el escritorio el archivo ejecutable de Terraform, que bajaremos del sitio oficial y estudiaremos en clase, en el caso de no poder crear una cuenta en Microsoft Azure, es aceptable también que descarguen el ejecutable de Terraform en su PC Personal.

Aprenderemos en esta tarea a crear Infraestructura en la Nube de Amazon Web Services utilizando:

- ❖ Terraform
- ❖ Nuestra cuenta de equipo de AWS
- ❖ Visual Studio con el plugin de Terraform
- ❖ Una vez construido nuestro código en Terraform, debemos customizarlo con el nombre de la instancia en el código, el nombre de la instancia debe ser el nombre de nuestro equipo "CloudDevopsTeamX" y el código de terraform, en los archivos main.tf, debe cargarse en github, por el Dev Team.

Sprint 6:

Una vez completados cada uno de los entornos de la tarea anterior, será necesario que despluguemos de manera manual la aplicación que el profesor les suministro, en 1 servidor en AWS, utilizando EC2, Creando una Instancia de Windows Server 2019 t2.micro, que tenga instalado IIS o en su defecto, WAMP, para así disponer de un servidor Web, en él, simplemente copiaremos nuestro código en el web server y visitaremos localhost para identificar que la aplicación se encuentra en la nube en una Instancia.

Deberemos actualizar el status de esta tarea en Trello y también en Github.

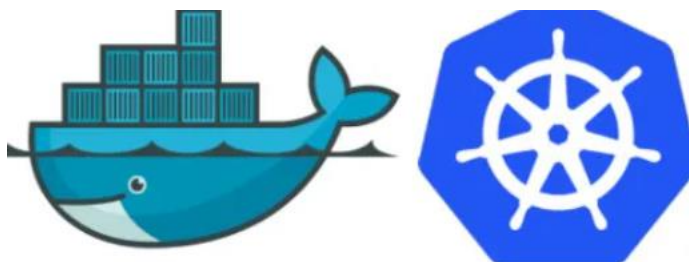
La instancia debe ser creada con acceso RDP desde cualquier dirección IP, también debe tener un disco EBS de 30gb, con un disco de SSD de 3era Generación, y en sus tags debe decir:

- ❖ Tipo: InstanciaWebServer
- ❖ OS: Windows Server2019
- ❖ Equipo: CloudDevopsTeamX
- ❖ Carrera: CloudDevOps
- ❖ Institución: EducacionIT
- ❖ Environment: Pre-Production

Una vez finalizado esta actividad, deberá ser reflejado en capturas en la tarea en el tablero de trello y también en Github

Sprint 7:

En esta sección nos dedicaremos a aprender Jenkins utilizando:



Jenkins en una Instancia en la Nube utilizando EC2, aquí crearemos una Instancia en AWS utilizando EC2 o en GCP utilizando Compute Engine, utilizando siempre las instancias que son de la capa gratuita, el profesor instruirá como hacerlo. En ellas, desplegaremos automáticamente Instancias que tendrán instalado Jenkins, y aprenderemos como crear un pipeline y cargaremos la aplicación que el profesor adjunto para aprender cada una de las fases del pipeline de Jenkins.

El detalle aquí es que la instancia debe tener los tags:

- ❖ Tipo: Windows Server con Jenkins
- ❖ Objetivo: CI-CD
- ❖ Carrera: CloudDevops
- ❖ Institución: EducacionIT

Una vez realizada la creación de la Instancia con Jenkins y realizado el laboratorio de los Pipelines de Jenkins, deberá cargarse capturas de la misma en Trello y en Github.

Sprint 8:

Aquí comenzaremos el estudio de Docker, para estas tareas será necesario:

- ❖ Que todos los miembros del Dev Team, tengan instalado Docker en Ubuntu. (Kali Linux también es una elección correcta)
- ❖ Todos los miembros del DevTeam tengan instalado Docker Composer
- ❖ Todos los miembros del DevTeam tengan una cuenta en DockerHub
- ❖ Todos los miembros del DevTeam tengan el código fuente de la aplicación que el profesor diseñó para el curso.
- ❖ Todos los miembros del DevTeam, deberán sacar capturas de tener instalado cada uno de los componentes solicitados y también adjuntar en Github los avances

Aquí aprenderemos a construir nuestra aplicación con Docker build, deberemos nombrar con `--name` a nuestra aplicación con el nombre de nuestro equipo "CloudDevopsTeamX" y sacar capturas de la aplicación ejecutándose en Docker con el nombre del equipo, esta captura debe adjuntarse en Trello y en Github.

Aprenderemos en los laboratorios de Docker como trabajar con la plataforma. Aquí no será necesaria una instancia de evaluación.

Sprint 9:

Será el momento de Trabajar con Kubernetes y aproximarnos al final de la carrera, aquí será necesario que todos los miembros del DevTeam tengan:

- ❖ Instalado Kubernetes en Ubuntu o Kali Linux
- ❖ Instalado Minikube en Ubuntu o Kali Linux
- ❖ Instalado VirtualBox en Ubuntu o Kali Linux
- ❖ Tengan una cuenta en AWS, GCP O Microsoft Azure
- ❖ Dispondrán de las guías de Minikube y Kubernetes y las guías de creación de Clusters en Amazon Kubernetes Service, Azure Kubernetes Service y Google Kubernetes Engine

Sprint 10:

Objetivos Finales.

- ❖ Con la aplicación que el profesor nos envió, debemos exponerla como un servicio en Kubernetes utilizando minikube y creando clusters con Minikube y Kubernetes en Kali Linux o Ubuntu
- ❖ Una vez realizada esta tarea, debemos sacar capturas del servicio expuesto con el nombre de nuestro equipo "CloudDevopsTeamX"
- ❖ Actualizar Trello y Github
- ❖ Luego, deberemos crear las estructuras de Clusters y Nodos con Google Kubernetes Engine, Amazon Kubernetes Services y Azure Kubernetes Services,
- ❖ El profesor les suministrara las guías y las aplicaciones de prueba, es importante que nombren al clúster y a los nodos con el nombre de su Equipo: CloudDevopsTeamX
- ❖ Una vez finalizado, cada una de las actividades de cada una de las tareas, se dará por finalizada la cursada y aprobada la carrera.

Estaré atento para cualquier consulta.

Gino

gino.luciano.rojo@educacionit.com

gino.luciano.workcontact@gmail.com (laboral, no comunicarse salvo previo aviso)