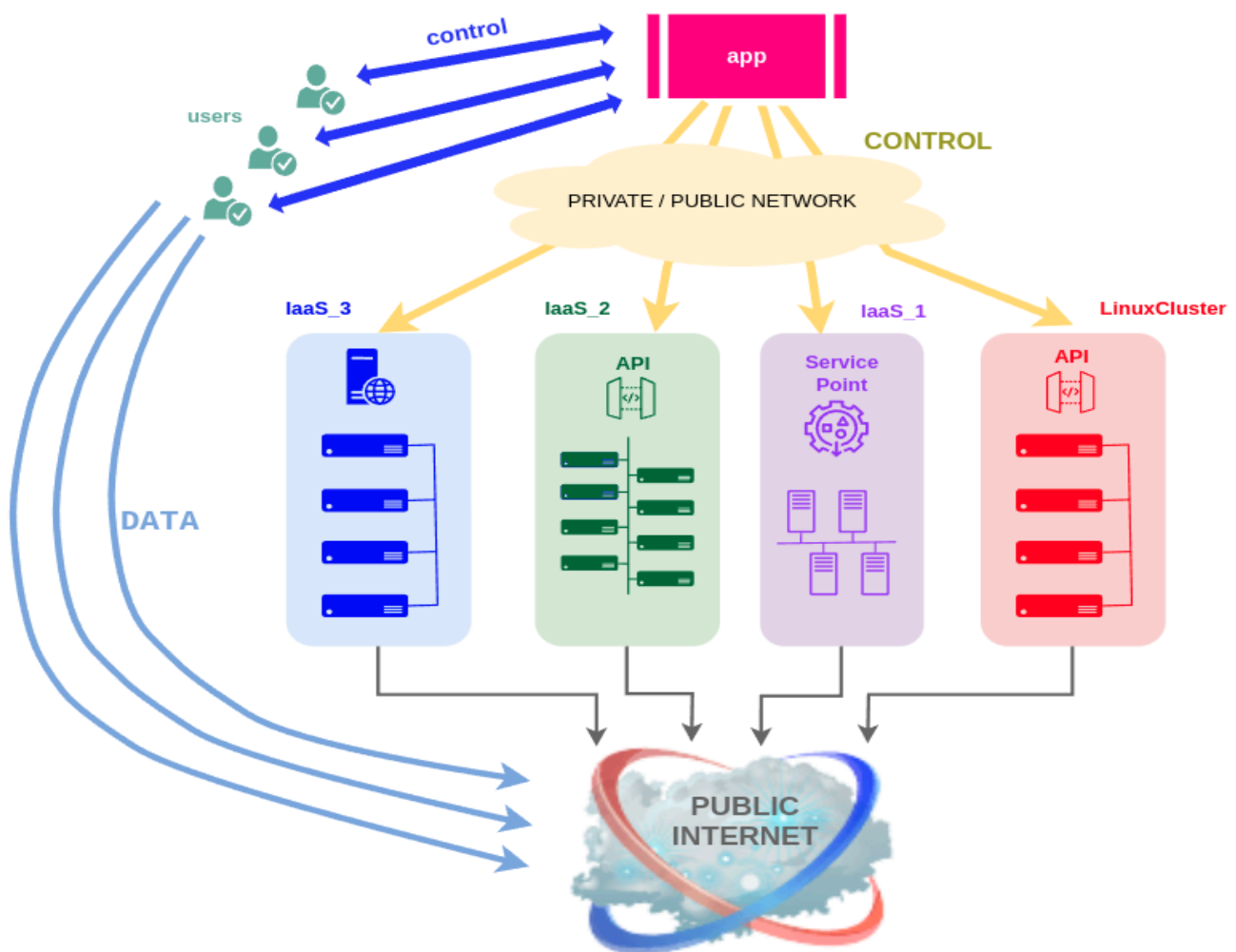


PROYECTO: PUCP Private Cloud Orchestrator

2025-1

La Pontificia Universidad Católica del Perú cuenta con una serie de servidores DELL Power Edge R720, los cuales planea usar para implementar una nube privada, de manera que pueda ser utilizada para el despliegue de topologías de máquinas virtuales conectadas dinámicamente en capa 2. Dada la cantidad de máquinas virtuales a provisionar, y el dinamismo, se requiere un **sistema automatizado** que ayude al administrador de la infraestructura con el despliegue y/o administración de los recursos.

Se le pide que implemente la primera versión de este sistema, la cual no tendrá un entorno gráfico. Su equipo de trabajo debe diseñar una solución que facilite el aprovisionamiento dinámico y minimice la posibilidad de error humano. La visión que se tiene para este proyecto se resume en la siguiente imagen:



Tal y como se observa en la imagen anterior, se busca tener una solución que nos permita manejar el ciclo de vida de topologías de red en diversos IaaS de manera particular (cada una de ellas puede tener un modo diferente de interacción). El usuario podrá acceder a la solución, la cual se comunicará con los IaaS para poder realizar las tareas que el usuario tenga disponible, dependiendo del rol que posea.

Es importante notar que el proyecto se realizará por etapas: (1) Diseño e implementación de orquestador con capacidades de manejo de ciclo de vida y uso de VMs usando tecnologías LINUX, (2) Aumento de funcionalidad para que el mismo orquestador pueda manejar el ciclo de vida y uso de VMs usando OpenStack. Es decir, al final del curso su orquestador deberá poder controlar el ciclo de vida de dos clusters (OpenStack y Linux) de una manera dinámica y accesible para el usuario.

Los requerimientos de esta solución son:

- **R0 [22 pts]:** Este requisito calificará la pulcritud de la exposición y la documentación, además de la cohesión y buena organización del grupo. Finalmente, la valoración personal y grupal que el coach realice será un factor determinante para este puntaje en específico (durante todo el ciclo). Para fines de calificación, la primera parte de R0 se aplicará a la documentación de arquitectura y demás parámetros de calificación mencionados líneas arriba. Finalmente para la segunda parte de R0 se aplicará a la documentación de la implementación.
- **R1 [40 pts]:** Controlar la creación y el borrado de slices de máquinas virtuales en una infraestructura de nube genérica, ya sea privada (Linux cluster, OpenStack, etc.) o pública (AWS, GCP, etc.).
- **R1B [18 pts]:** Proveer una interfaz de usuario que permita a un usuario gestionar el ciclo de vida de sus slices (creación, edición, borrado, etc.).
- **R1C [18 pts]:** Gestionar el despliegue/aprovisionamiento de slices sobre la Infraestructura de nube deseada. Responsable porque los diferentes elementos del slice sean creados en el orden adecuado y en el lugar deseado y que la creación del slice sea completa (no se quede a medias).
- **R2 [18 pts]:** Soportar la creación/borrado de slices sobre un cluster de servidores Linux.
- **R3 [30 pts]:** Soportar la creación/borrado de slices sobre un sistema OpenStack.
- **R4 [30 pts]:** Ejecutar el "VM Placement" (mapeo de VM a servidor físico) tomando en cuenta los recursos disponibles y permitiendo elegir la "zona de disponibilidad" a usar.

- **R5 [24 pts]:** Proveer soporte de Networking capas 2 y 3, permitiendo definir reglas de ciberseguridad

Su grupo diseñará e implementará una solución que cubra tantos requerimientos como les sea posible. En función a la calidad de su solución, recibirá un puntaje total (nota) que será repartido entre sus miembros de la manera que los miembros del grupo decidan.

Adicionalmente, como parte del requerimiento cinco (R5) se espera que las VMs desplegadas en su solución puedan ser accedidas desde el “exterior”, para esto se han definido rangos de IPs por grupos, las cuales serán seteadas en sus dispositivos (pueden usar DHCP). Ahora, no necesariamente todas las VMs deben ser vistas desde el exterior, en realidad, sólo las que ustedes explícitamente decidan.

GRUPO	LINUX CLUSTER	OPENSTACK CLUSTER
1	10.10.1.0/24	10.10.2.0/24
2	10.10.3.0/24	10.10.4.0/24
3	10.10.5.0/24	10.10.6.0/24
4	10.10.7.0/24	10.10.8.0/24
5	10.10.9.0/24	10.10.10.0/24
6	10.10.11.0/24	10.10.12.0/24

Como pueden notar, las IPs contenidas en el cuadro anterior, son **IPs PRIVADAS**, las cuales van a poder ser accedidas **sólo** a través de la VPN. Si por algún motivo intentamos acceder a ellas desde dentro de la UNIVERSIDAD obtendrán errores de conexión, ya que no están disponibles a nivel INTERNO (Universidad), pero **sí** a través de la **VPN**, ya que se configurarán rutas para que puedan permitir dichas conexiones usando como router al Headnode del VNRT.

Criterios de evaluación

Descripción	Criterios
<p>R0</p> <p>Performance en presentación, temas requeridos y documentación.</p> <p>(22 pts)</p>	<ul style="list-style-type: none"> • Claridad de la exposición y manejo de medios. • Cohesión y trabajo en equipo durante el periodo a evaluar. • Desempeño personal constante durante el proyecto. • Orden, redacción, contenido y buena organización de la presentación. • Porcentaje de Cobertura de los requerimientos.
<p>R1</p> <p>Controlar la creación y el borrado de slices de VMs en una infraestructura de nube privada.</p> <p>(40 pts)</p>	<ul style="list-style-type: none"> • Diseño independiente de la capa de infraestructura (cluster de servidores Linux, OpenStack, AWS, etc.). • Debe tener un módulo para cada uno de los requerimientos subsiguientes: <ul style="list-style-type: none"> - Debe poder almacenar la información de forma persistente. - Debe permitir la comunicación en red entre módulos. - Debe soportar la creación concurrente (multiusuario) de slices, es decir, manejo de colas. - Debe soportar la creación de logs con la información necesaria para hacer troubleshooting de cada módulo.
<p>R1B</p> <p>Interfaz de usuario (UI) que gestiona el ciclo de vida de slices</p> <p>(18 pts)</p>	<ul style="list-style-type: none"> • Permitir que múltiples usuarios usen la infraestructura de forma concurrente. • Permitir seleccionar slices con topologías predefinidas: lineal, malla, árbol, anillo, y bus. • Permitir <u>editar slices existentes</u> agregando nodos y enlaces. • Permitir especificar la configuración (capacidad) de cada VM a crear. • Listar los slices existentes. • Listar el consumo de recursos del sistema. • Permitir borrar slices previamente creados. • Debe permitir importar imágenes de VMs. • Proveer tokens/credenciales para que el dueño del slice pueda acceder a la consola virtual de cada VM.
<p>R1C</p> <p>Gestor de despliegue de slices ("Slice Manager")</p> <p>(18 pts)</p>	<ul style="list-style-type: none"> • Expone los APIs del sistema. Puede ser un sólo módulo o varios. • La comunicación del resto de módulos debe pasar por él. • Debe validar que el usuario esté autenticado y tenga autorización de realizar la acción solicitada. • Debe validar que el body de las solicitudes tengan la sintaxis correcta. • Modela los slices y sus elementos (uso de OOP). • Crea los elementos en el orden correcto y lidia con errores y con conflictos por el uso de recursos.

<p>R2</p> <p>Soporte de cluster de servidores Linux</p> <p>(28 pts)</p>	<ul style="list-style-type: none"> • Orquesta VM sobre servidores corriendo alguna versión de linux. • Ahorro de almacenamiento: debe borrar las imágenes de las VMs que ya no se usen (en los servidores). • Proveer acceso a la consola virtual de cada VM.
<p>R3</p> <p>Soporte de OpenStack</p> <p>(20 pts)</p>	<ul style="list-style-type: none"> • Usar el OpenStack API. • Implementar al menos Keystone, Horizon, Nova, Neutron (ML2 plugin), y Glance. • Debe permitir ver cada slice creada, por separado, en Horizon.
<p>R4</p> <p>VM Placement</p> <p>(20 pts)</p>	<ul style="list-style-type: none"> • Asignar las máquinas virtuales a servidores de acuerdo a su criterio de performance. Por ej: <ul style="list-style-type: none"> - El consumo de recursos sea aceptable de acuerdo a su SLA (p.ej., no exceda un nivel máximo de sobre-aprovisionamiento, el cual sería un parámetro de configuración). - La carga está balanceada entre los servidores. - Se minimice el consumo de energía - Se maximiza la performance del slice, visto como un todo. • Listar los recursos (servidores) disponibles. • Permitir al usuario definir zonas de disponibilidad. • Permitir al usuario elegir en qué zona de disponibilidad se desplegará un slice.
<p>R5</p> <p>Networking y Seguridad</p> <p>(22 pts)</p>	<ul style="list-style-type: none"> • Proveer salida a Internet y reglas de seguridad a las VMs que lo requieran. • Debe permitir acceso desde Internet a las máquinas virtuales que lo requieran (equivalente a tener una IP pública). • Usar redes provider (VLANs) en lugar de self-service (tuneles). • Limitar los paquetes de broadcast a aquellos puertos/servidores que tienen VMs que pertenezcan a la subred respectiva (ya sea usando Netcong, Yang, OpenFlow, etc.).

Plan de acción

Sem	Fecha	Actividades
1	24 Mar / 30 Mar	<ul style="list-style-type: none"> - Repaso Linux, Bash, Python - Revisión del diplomado en Cloud Computing & otra bibliografía.
2	31 Mar / 6 Abr	<ul style="list-style-type: none"> - Repaso de Linux Networking y Linux Process Management - Lab 1: Bash & Python scripting + Linux Process Management - Primera exposición y discusión del proyecto y sus alcances.
3	07 Abr / 13 Abr	<ul style="list-style-type: none"> - Revisión del diplomado en Cloud Computing & otra bibliografía - Revisión de tesis de Ruben Cordova y Antonio Merino - Lecturas adicionales provistas por "coach" y los miembros. - Redacción de requerimientos del proyecto
4	14 Abr / 20 Abr	<ul style="list-style-type: none"> - Repaso de Tecnologías de Virtualización - Lab 2: Aprovisionamiento de un slice de VMs en un solo host. <p>Presentar, como parte del reporte final del lab 2:</p> <ol style="list-style-type: none"> 1. concepto de operación y/o casos de uso 2. descripción de entidades y roles en su solución 3. bosquejo de arquitectura (módulos e interacciones entre ellos).
5	21 Abr / 27 Abr	<ul style="list-style-type: none"> - Repaso de Tecnologías de Virtualización (cont.) <p>Presentar:</p> <ol style="list-style-type: none"> 1. Desarrollo del borrador de arquitectura de solución: módulos y sus funciones.
6	28 Abr / 04 May	<ul style="list-style-type: none"> - Repaso de Tecnologías de Virtualización (cont.) - Lab 3: Avances de la arquitectura proyecto + aprovisionamiento de Slices en múltiples hosts LINUX (parte 1) - FEEDBACK DEL PROFESOR <p>Presentar:</p> <ol style="list-style-type: none"> 1. El documento y PPT con desarrollo de arquitectura debe subirse a PAIDEIA 1 día antes del Lab 3.
7	05 May / 11 May	<ol style="list-style-type: none"> 1. Repaso de Tecnologías de Virtualización (cont.) <p>Presentar:</p> <ol style="list-style-type: none"> 2. Primera versión de código para despliegue de slices(lineal y anillo) en múltiples hosts LINUX (sin APIS, sólo funciones).
8	12 May / 18 May	<ol style="list-style-type: none"> 1. Repaso del problema de "VM Placement" 2. Lab 4: Avances de arquitectura del proyecto + aprovisionamiento de un slice de VMs en múltiples hosts LINUX (parte 2) 3. FEEDBACK DEL PROFESOR <p>Presentar:</p> <ol style="list-style-type: none"> 4. El documento con HLD (R2 - Soporte para LINUX CLUSTER) debe subirse a paideia 1 día antes del Lab 4 (parte de nota del lab) 5. Exposición en el laboratorio del documento presentado.

9	19 May / 25 May	<ul style="list-style-type: none"> - 1^{RA} PRESENTACIÓN PROYECTO: - ARQUITECTURA GENERAL DEL PROYECTO - DISEÑO HIGH LEVEL DE LOS MÓDULOS R1, R1B y R1C, R2, y R5. - LA IMPLEMENTACIÓN DEL MÓDULO R2(Soporte de Linux Cluster). <p>El módulo R2 deberá interactuar con el driver(SDK) que hayan definido como parte de sus módulos y para probar el funcionamiento deberán desplegar la siguiente topología (EX1-Topology), además de las pre-definidas (lineal y anillo).</p>
10	26 May / 01 Jun	<ul style="list-style-type: none"> - Repaso del problema de "VM Placement" (cont.) <p>Presentar:</p> <ol style="list-style-type: none"> 1. Desarrollo del High level software design de los módulos R3 y R4.
11	02 Jun / 08 Jun	<ul style="list-style-type: none"> - Repaso de OpenStack - Lab 5: Análisis y despliegue de OpenStack <p>Presentar</p> <ol style="list-style-type: none"> 1. Implementación completa del User Interface (R1B).
12	09 Jun / 15 Jun	<ul style="list-style-type: none"> - Repaso de OpenStack (cont.) - Revisión de tesis de Gabriel Gamero. - Definición con coaches del plan de pruebas. - Implementación completa de los requerimientos R1C y R5. - Unit testing: la aplicación imprime el output de <u>todos</u> los comandos que enviaría al resto de módulos pendientes de implementación
13	16 Jun / 22 Jun	<ul style="list-style-type: none"> - Repaso de Contenedores - Lab 6: - FEEDBACK DEL PROFESOR <ul style="list-style-type: none"> o Informe Previo: Aprovisionamiento de VMs usando: (i) la CLI de OpenStack , y (ii) la API de OpenStack. o Experiencia de Lab: presentación de la implementación, el plan de pruebas y los resultados de las pruebas realizadas hasta el momento. <p>Presentar:</p> <ol style="list-style-type: none"> 1. Un(1) día antes del Lab 6 deben subir el plan de pruebas.
14	23 Jun / 29 Jun	<ul style="list-style-type: none"> - Repaso de Contenedores (cont.) - Completar la implementación de los módulos R3 (despliegue de VMs en OpenStack) y R4 (algoritmo de VM Placement).
15	30 Jun / 06 Jul	<ul style="list-style-type: none"> - El mercado de Cloud Computing - Lab 7: - FEEDBACK DEL PROFESOR - Implementación de un módulo del proyecto del curso, incluye UNIT Testing y reporte de pruebas.
16	07 Jul / 13 Jul	<ul style="list-style-type: none"> - PRESENTACIÓN FINAL

Entregables

Primera Entrega	1. Arquitectura de la solución (documento PDF, requerimiento R1) incluyendo APIs y/o IPC (arquitectura del resto de requerimientos). Debe indicar cómo la arquitectura permite satisfacer sus requerimientos.
	2. High level software design (documento PDF) de los requerimientos R1B, R1C, R2, y R5(incluido R0 primera parte).
	3. Copia de presentación (powerpoint, PDF, etc.)
	4. Código actual, incluyendo especialmente el código relacionado al requerimiento R2
	5. Durante el examen parcial: Demo del despliegue de slide predefinida (req. R2).
Entrega Final	6. High level software design (documento PDF) de todos los módulos. Presentar en un mismo documento con la arquitectura (actualizada de ser necesario)
	7. Plan de pruebas y reporte de plan de pruebas.
	8. Copia de presentación (powerpoint, PDF, etc.)
	9. Código de la versión final de su prototipo.
	10. Durante examen final: Demostración de funciones a pedido de los jurados.

ASESORÍAS

Horario	Grupo	Coach	Reunión semanal	Miembros
991	G1	Branko Zambrano	Miércoles 8PM	Juan Alfonso Chapoñan (L) Sergio Sotomayor Ana Paula Gonzalez
	G2	Jenry Luis	Lunes 10PM	Pablo Flores (L) Gerardo Gutierrez Aracelli La Rosa Mayte Asto
	G3	José García	Domingo 7PM	Manuel Aliaga (L) Piero Fernandez Alexander Abrisqueta Jefferson Villegas
	G4	Ronny Pastor	Lunes 8PM	Dana Nolasco (L) Andrea Neira Francis Estela
992	G5	Alejandro Macedo	Viernes 8PM	Damian Lopez (L) Juan Loyola Leonardo Abanto Sebastian Segura
	G6	Branko Zambrano	Jueves 8PM	Hineill Cespedes (L) Gabriel Talledo Denilson Quispe Sergio Bustamante

Rol de Presentaciones / Retroalimentación profesor

	Fecha	Grupos
Concepto de Operación	Martes 08 Abril, 4:00PM (Clase sem 3)	G3, G6
Casos de Uso	Martes 15 Abril, 4:00PM (Clase sem 4)	G4, G1
Bosquejo de arquitectura	Martes 22 Abril, 4:00PM (Clase sem 5)	G5, G2
1^{ra} presentación (Arquitectura, etc.)	Martes 29 Abril, 8:00PM (Lab 3 H0991)	G1 y G4
	Martes 06 Mayo, 4:00PM (Clase sem 7)	G2 y G3
	Jueves 15 Mayo, 5:00PM (Lab 4 H0992)	G5 y G6
Algoritmo de VM Placement & estrategia de monitoreo de recursos	Martes 03 Junio, 4:00PM (Clase sem 11)	G3, G1
Configuración de OpenStack cluster (R3)	Martes 10 Junio, 4:00PM (Clase sem 12)	G6, G4
2^{da} presentación (Plan de pruebas, etc.)	Martes 17 Junio, 8:00PM (Lab 6 H0991)	G2 y G3
	Jueves 19 Junio, 5:00PM (Lab 6 H0992)	G5 y G6
	Martes 24 Junio, 4:00PM (Clase sem 14)	G1 y G4

Calificación del curso

La nota final del curso se calculará con la siguiente fórmula:

$$Nf = 0.4Pb + 0.3Ex1 + 0.3Ex2$$

Donde

Pb : Promedio de los 7 laboratorios.

Ex1 : Nota asignada a la primera entrega del proyecto.

Ex2 : Nota asignada a la segunda (y final) entrega del proyecto.

A continuación se muestra una tabla con la distribución de puntajes

	Requerimientos								
	R0	R1	R1B	R1C	R2	R3	R4	R5	
Arq / Doc	8	40	6	6	6	12	12	10	100
Imp	14	N/A	12	12	12	18	18	14	100
	22	40	18	18	18	30	30	24	200

Nota de EX1: corresponde a los requerimientos R0, R1, R1B y R1C, R5 (más el componente de arquitectura y documentación para R0), más la componente de arquitectura / implementación del módulo R2. Total 100 pts.

Rúbrica de calificación: [EX1](#)

Script de validación: [script](#)

Nota de EX2: corresponde al componente de implementación de los requerimientos R1B,R1C; además de R0(documentación) y finalmente R5 más el componente de diseño/arquitectura de los requerimientos R3 y R4. Total 100 pts.

En ambas presentaciones, basta obtener 20*(número de integrantes) de los 100 puntos para ser asignado el puntaje máximo.