

حل امتحان (2025) Cloud Computing

تمرين 1 (6) : الإجابات الصحيحة

من فوائد الـ virtualisation : <input type="checkbox"/> تحويل الموارد إلى رقمية <input type="checkbox"/> الإدارة المركزية × <input type="checkbox"/> دمج الموارد	الحوسبة الشبكية Grid Computing مخصصة لحل مشاكل : <input type="checkbox"/> الوقت <input type="checkbox"/> التخزين × الحسابات الموزعة
الجانب الاقتصادي للحوسبة السحابية : <input type="checkbox"/> مبني على اتفاقية (SLA) <input type="checkbox"/> مرونة سريعة × <input type="checkbox"/> الدفع حسب الاستخدام	الجانب التقني للحوسبة السحابية : <input type="checkbox"/> التوافرية × <input type="checkbox"/> مبني على المحاكاة × خدمة مقاسة
ما هي ميزة الـ Multi-tenancy في Cloud ؟ <input type="checkbox"/> استخدام عدة حسابات × <input type="checkbox"/> استخدام بيئة واحدة لخدمة عدة عملاء معزولين <input type="checkbox"/> تقليل الأمان بين المستخدمين <input type="checkbox"/> نسخ البيانات تلقائياً	لماذا نستخدم الحاويات containers ؟ <input type="checkbox"/> أقل كفاءة من الآلات الافتراضية × إمكانية النقل بين البيئات <input type="checkbox"/> فصل بيئات التطوير والاختبار والإنتاج
أنواع التخزين في السحابة تشمل <input type="checkbox"/> Block Storage × <input type="checkbox"/> Object Storage <input type="checkbox"/> تخزين الصور	أي من هذه النماذج تمثل مزيجاً بين السحابة العامة والخاصة؟ <input type="checkbox"/> Public Cloud <input type="checkbox"/> Hybrid Cloud × <input type="checkbox"/> Duo-cloud
مثال على خدمة IaaS : <input type="checkbox"/> EC2 × <input type="checkbox"/> Elastic Beanstalk <input type="checkbox"/> Lambda <input type="checkbox"/> Comput Engine	عرض RDS هو من النوع : <input type="checkbox"/> PaaS × <input type="checkbox"/> SaaS <input type="checkbox"/> IaaS
عرض EC2 هو من النوع : <input type="checkbox"/> PaaS <input type="checkbox"/> SaaS <input type="checkbox"/> IaaS ×	ما وظيفة CloudSim في Cloudlet ؟ <input type="checkbox"/> تمثيل المستخدم <input type="checkbox"/> محاكاة مركز بيانات × محاكاة مهمة أو وحدة عمل
ما المقصود بـ Auto-scaling في الحوسبة السحابية ؟ <input type="checkbox"/> ميزة لإغلاق الخوادم تلقائياً <input type="checkbox"/> نظام لتحديث التطبيقات <input type="checkbox"/> أداة لحماية البيانات <input type="checkbox"/> تقنية لزيادة أو تقليل الموارد تلقائياً حسب الطلب	ما هي الـ SLA في Cloud ؟ <input type="checkbox"/> Service Level Agreement × <input type="checkbox"/> Secure Layer Agreement <input type="checkbox"/> System Logic Access <input type="checkbox"/> Software License Agreement
لتنصيب OpenStack : <input type="checkbox"/> Keystone <input type="checkbox"/> Novastack <input type="checkbox"/> Microstack × <input type="checkbox"/> Devstack	من مزودي خدمات الحوسبة السحابية : <input type="checkbox"/> Docker hub <input type="checkbox"/> Youtube <input type="checkbox"/> Infomaniak ×
Orchestration : <input type="checkbox"/> Docker <input type="checkbox"/> OpenStack <input type="checkbox"/> Cloudsim	خدمات من النوع XaaS : <input type="checkbox"/> Backup aas × <input type="checkbox"/> Frontend aas <input type="checkbox"/> Backend as a service
عنصر خدمة المصادقة (authentication) : <input type="checkbox"/> Neutron <input type="checkbox"/> Swift <input type="checkbox"/> Glance	نماذج التسعير لـ EC2 : <input type="checkbox"/> Revised <input type="checkbox"/> Commercial instance <input type="checkbox"/> Used

التمرين 2 (5) :

Cloud Broker يعمل كوسيط بين مزودات الخدمات السحابية (مثل AWS, Azure, ...) والمستخدمين. هدفه: تبسيط، وتحسين، وتنسيق استخدام الخدمات السحابية.	Full Virtualization هي تقنية افتراضية يتم فيها محاكاة العتاد بالكامل بحيث يمكن تشغيل أي نظام تشغيل ضيف دون تعديلات فوق نظام مضيف ، Para-virtualisation هي تقنية يُعدل فيها نظام التشغيل الضيف ليتفاعل مباشرة وبشكل دقيق مع الـ Hypervisor ، دون الحاجة لمحاكاة كاملة للعتاد ودون الحاجة لنظام مضيف.
VMware ESXi هو Hypervisor من النوع Para-virtualisation VMware vSphere هو الحل الكامل الذي يستخدم ESXi مع أدوات إدارة قوية لبناء بيئة افتراضية سحابية.	SaaS – البرمجيات كخدمة هو نموذج تقدم فيه الشركات تطبيقات جاهزة للاستخدام عبر الإنترنت. – PaaS المنصة كخدمة هو نموذج يوفر للمطورين بيئة جاهزة للتطوير وتشغيل ونشر التطبيقات دون القلق بشأن إدارة الخوادم أو أنظمة التشغيل.
Docker هو أداة لإنشاء وتشغيل حاوية واحدة أو عدة حاويات على جهازك . Kubernetes هو نظام تنسيق (Orchestration System) لإدارة الحاويات على نطاق واسع. يعمل فوق Docker (أو أي Container Runtime) لإدارة مئات أو آلاف الحاويات تلقائياً.	OpenStack هو منصة مفتوحة المصدر لإنشاء ونشر وإدارة بنية تحتية سحابية حقيقية (IaaS). تُستخدم لإنشاء سحابة خاصة أو عامة، وتدير موارد مثل: الخوادم، التخزين، الشبكة... CloudSim هو أداة محاكاة (Simulation Tool) مبنية بلغة Java، تُستخدم لمحاكاة بيئة الحوسبة السحابية.
Cloud Agility هي قدرة الشركة أو النظام على الاستجابة السريعة للتغيرات في السوق أو الاحتياجات التقنية، باستخدام موارد الحوسبة السحابية. أما Elasticity هي قدرة النظام على زيادة أو تقليل الموارد تلقائياً حسب حجم الطلب (توسّع وانكماش تلقائي).	Neutron هو خدمة الشبكات في OpenStack و Glance هو خدمة إدارة صور الأجهزة (VM images)
Elastic Beanstalk هو خدمة PaaS من AWS تتيح للمطور نشر التطبيقات بسرعة دون القلق بشأن البنية التحتية. (Compute Engine هو خدمة IaaS (البنية التحتية كخدمة) من Google Cloud . تتيح إنشاء وإدارة خوادم افتراضية (VMs) مخصصة بالكامل.	الصورة هي قالب ثابت يحتوي على كل ما يحتاجه التطبيق ليعمل. غير قابلة للتغيير. تُستخدم كنقطة انطلاق لإنشاء حاويات. يمكن تخزينها في مستودعات مثل Docker Hub. اما Container حاوية هي مثل (Instance) يتم تشغيله من Image .

التمرين 3 (4) : اشرح بالتفصيل ما يظهر في البرنامج التالي؟ يتم إنشاء بيئة سحابية بمحاكي CloudSim.

<p>السطر لتهيئة محرك المحاكاة. مثلا 1: عدد المستخدمين إنشاء مركز بيانات ويستخدم TimeShared للجدولة إنشاء وسيط مسؤول عن إدارة الـ VMs و Cloudlets . يتم إنشاء VM 2 بنفس الخصائص MIPS: 1000 سرعة المعالج RAM: 512 MB ... جدولة TimeShared للمهمة Cloudlet أي يتم تقاسم وقت التنفيذ بين Cloudlets داخل VM يُبلغ الـ Broker بقائمة الآلات الافتراضية التي أنشأها. يتم إنشاء 4 Cloudlets مهام: (length: طول المهمة = fileSize/outputSize: عدد التعليمات (40000 MI لكل منهما UtilizationModelFull: تعني أن المهمة تستهلك 100% من الموارد طوال فترة التنفيذ. يتم توزيع الـ Cloudlets على الـ VMs باستخدام القسمة % i (أي: مهمة 0 و 2 على VM0، مهمة 1 و 3 على VM1)</p> <p>يُبلغ الـ Broker بالمهام التي يجب تنفيذها على الـ VMs يبدأ تنفيذ جميع الأحداث المجدولة ضمن بيئة المحاكاة. يتم الحصول على قائمة المهام التي اكتملت مع بياناتها. يتم إيقاف المحاكاة وطباعة تفاصيل نتائج تنفيذ كل Cloudlet.</p>	<pre>CloudSim.init(1, Calendar.getInstance(),false); Datacenter datacenter = createDatacenter("Datacenter_1", "TimeShared"); DatacenterBroker broker = new DatacenterBroker("Broker"); List<Vm> vmList = new ArrayList<> ();int vmid=0;int mips=1000;long size=10000; int ram=512; long bw = 1000;int pesNumber = 1; String vmm = "Xen;" for (int i = 0; i < 2; i++) { vmList.add(new Vm(vmid++, broker.getId(), mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())); } broker.submitVmList(vmList); List<Cloudlet> cloudletList = new ArrayList<>() ; int id =0;long length=40000;long fileSize=300;long outputSize = 300; UtilizationModel utilizationModel = new UtilizationModelFull(); for (int i = 0; i < 4; i++){ Cloudlet cloudlet = new Cloudlet(id++, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationModel, utilizationModel); cloudlet.setUserId(broker.getId()); cloudlet.setVmId(vmList.get(i).% vmList.size()).getId()); cloudletList.add(cloudlet); } broker.submitCloudletList(cloudletList); CloudSim.startSimulation(); List<Cloudlet> newList = broker.getCloudletReceived(); CloudSim.stopSimulation();</pre> <div data-bbox="901 526 1524 672"> <p>يتم إنشاء بيئة سحابية بمحاكي CloudSim</p> <ul style="list-style-type: none"> • يتم إنشاء 2 VM و 4 مهام. • يتم جدولة وتنفيذ المهام على الـ VMs باستخدام TimeShared • يتم جمع نتائج التنفيذ. </div>
--	--

التمرين 4 (5) :

<p>(2) لنشر (Deploy) تطبيق ويب في OpenStack نستخدم لوحة التحكم Horizon Dashboard أو أدوات السطر مثل OpenStack CLI المتطلبات الأساسية</p> <ul style="list-style-type: none"> • حساب مستخدم في OpenStack • صورة نظام تشغيل مثل Ubuntu • إعدادات شبكة (Network) • مفتاح SSH للدخول إلى الملقم لاحقاً. • تطبيق الويب جاهز للنشر. • الخطوات التفصيلية للنشر عبر Horizon Dashboard <p>(1) تسجيل الدخول</p> <p>(2) إنشاء زوج مفاتيح (Key Pair)</p> <p>(3) إعداد الشبكة : أنشئ شبكة (Network) ، Subnet، ثم Router يربطها بالإنترنت</p> <p>(4) إطلاق آلة افتراضية (VM) : من Project > Compute > Instances اضغط Launch Instance.</p> <ul style="list-style-type: none"> • الوصول إلى الـ VM : بعد تشغيل الـ VM ، احصل على عنوان IP العمومي (Floating IP) . ثم استخدم SSH للدخول إلى الملقم <p>(5) تثبيت بيئة التشغيل</p> <p>(6) تحميل التطبيق وتشغيله : يمكنك استخدام scp أو git clone لجلب التطبيق:</p>	<p>(1) تمثل بنية لخدمة Virtual Private Cloud سحابة خاصة افتراضية عند AWS به Web Servers في Public Subnet عبر الإنترنت و قاعدتان للبيانات لا يمكن الوصول إليها مباشرة من الإنترنت.</p> <ul style="list-style-type: none"> • Region: تمثل منطقة جغرافية تحتوي على موارد AWS • Availability Zone : منطقة فرعية داخل الـ Region • VPC (Virtual Private Cloud) : سحابة خاصة افتراضية تُستخدم لتوزيع الموارد (سيرفرات...) • Public Subnet : جزء من الشبكة يسمح بالوصول إلى الإنترنت يحتوي على: <ul style="list-style-type: none"> – Web Servers مثلاً Apache – Security Group : جدار حماية يتحكم في حركة المرور القادمة والخارجة أي في من يمكنه: إرسال طلبات و استقبال الردود • NAT Gateway : تُستخدم للسماح للـ Private Subnet بالوصول إلى الإنترنت (مثل تحميل تحديثات)، لكن دون تعريضها للوصول المباشر من الخارج. • Private Subnets : شبكات فرعية لا يمكن الوصول إليها من الإنترنت مباشرة. تحتوي على قاعدتان للبيانات MySQL Database Servers و MongoDB Database Servers • Internet Gateway : البوابة التي تسمح للـ VPC بالاتصال بالإنترنت.
---	--