

Task 8 Answers



=> ايه هي مميزات ال **super harvard modified** ؟

- الي عملوه زياده عن ال modified انهم كبروا مساحة ال cache memory
- حطوا I/O controller

=> امتى نستخدم **pip** ومنستخدمش **conda** ؟

- لما يكون في library مش موجوده في Conda وموجوده في pip لكن في العموم conda احسن من pip لأنها وعلى الرغم انها أبطاء فهي بتوفر كل ال dependencies الي بتعتمد عليها ال library
- بس عيب conda انك لو نزلت حاجه ب version وجيت تنزلها ب version ثاني فهي هتنزل ال version 2 اما pip هيعمل override

=> ايه هي ال **solid principles** ؟

1-single responsibility وهو انه ال class يكون مسئول عن حاجه واحده بس وبالتالي لما يجي يتغير فهو هيتغير عشان سبب واحد بس

2-open closed principle وهو ان ال classes وال moduls وال functions لازم يكونوا open لل extension و closed لل modification يعني اقدر اعمل add لأي behaviour انا عاوزه اضيفه عليهم من غير ما اغير ال behaviour الاساسي بتاعهم

3-Liskov substitutions هو انه الابن يقدر يحل محل الأب لو غاب بمعنى عندنا class S و class T وهو child ل T فمغنى كده انه S يقدر يعمل اي حاجه بيعملها T فبالتالى لو جيت عملت كده (T t=new T) ده الطبيعى لكن كمان ممكن تعمل كده

T t = new S()

4-Interface segregation وهو ببساطه اني بقسم ال Interfaces الكبيره ل Interfaces صغيره عشان لو جينا نعمل لحاجه منهم implement ناخذ بس ال methods الي محتاجينها

5-Dependency inversion ان لو في high module بيعتمد على low module منخليهمش يعتمدوا على بعض عن طريق ال abstraction

=> ايه هو ال distributed system topology ؟

- ال topology بتاعت ال ds يعني ازاي الأجهزة دي متوصله ببعض او ايه هي طرق توصيل ال network مابينهم
- بنحدد اي topology على اي اساس ؟ على اساس ال system بتاعنا يعني في مثلا system عاوز سيرفر بيعت لكل الاجهزة ومفيش حاجه تتبعتله و system تاني محتاج انه السيرفر بيعت ويتبعتله وهكذا

=> ايه هو , dockerization, containerizations , kubernetes,

Jenkins؟ "الحوار كبير فكوبايتك وتعالى "



- خلونا نجيب الشريط من اوله ونتكلم ليه الحاجات دي ظهرت كان في مشكله زمان انه اغلب المشاريع الضخمه بتكون عباره عن مجموعه كبيره من ال tools وال packages وال libraries المعتمده على بعض والكود بتاع ال project بيكون معتمد عليها بشكل اساسي وده بيدخلنا في حاجه اسمها ال dependancy hell

- المشكلة اننا لما نيجي ننقل ال app بتاعنا من environment لمكان ثاني خالص بتضطر انك ت install كل ال packages وال libraries دي ثاني يعني مثلا خلصنا ال app بتعنا على machine شغاله windows مثلا فانت عاوز تجرب ال app على linux فتهتضر انك تسطب كل الحاجات الي ال app ده بيعتمد عليها على الجهاز الي فيه linux
- ال containerizations ك concept فكرته انه عندنا حاجة اسمها container بنجمع فيه كل ال packages وال libraries الي ال app بتاعنا هيحتاجها وهيوفرله ال enviroment كمان المناسبه ليه عشان ي run يعني يوم ما تحب انك تجرب ال app بتاعك في operating system ثاني او اي جهاز ثاني عموما ال container بيقولك ملكش دعوه اديني انت ال app وانا هعمل كل حاجة تخليه ي run
- والي بيطبق الفكره دي هو ال docker فمفهوم ال dockerization ما هو الا اننا بنطبق مفهوم ال containerizations باستخدام برنامج ال docker وهو عبارته عن open source cloud بترفع عليه ال app بتاعك وت run عليه
- في مفهوم ثاني ظهر قبل ال containerizations وهو ال virtualizations وده فكرة اني انزل os ثاني على نفس ال operating system الي انا شغاله عليه as virtual machine بدل ما يجبوا جهاز ثاني ب os مختلف لا هما هينزلوا وهيجربوا عليه ال app بس مشكلته انه بيستهلك ال RAM وال process بتبقى بطيئه جدا فال container ظهر بعد الفكره دي عشان يقدروا يقللوا من استهلاك ال ram وال processes متتاثرش
- خلصنا docker "قول الحمد لله " ندخل على kubernetes "قولتلك الحوار محتاج خربوش شاي "
- الفكره بتاعته جات منين ؟ هي جات لما جوجل كانت محتاجه تنظم ال containers بتاعتها وتجمعهم تحت سيطرة حاجه بتنظمهم وتنحكم فيهم كلهم وعملوا حاجه اسمها BORG بتعمل كده وبعدها ه عملوا حاجه ثاني افضل اسمها omega وبعدين جوجل قررت ت share الكلام ده مع العالم كله "اكيد مش لله والوطن " فاخترعوا kubernetes الي هو عبارته عن Container كبير في ال containers صغيره وال containers دي في حاجات مشتركه فيها فعشان كده اتخطوا في نفس ال container الكبير

- ندخل على Jenkins اخر حاجه معانا "تعبت ؟ قومت كملته
للي قرأها صح عاش ي سنجل يا بئس الناس الثانيه ربنا يغفرلكم "
- هي عباره عن open source مكتوب ب java بنستخدمها في عملية ال
continuous integration وال continuous delivery لما تيجي ترفع
الكود بتاعك على github اول ما بتعمله push بياخده يعمل build وبي
test الكود لو في اي خطأ فهو هيقف عملية ال building وهيلغي الحاجه
الي انت رافعته لحد ما تظبط الكود بتاعك ولما تصححه ويعدي من مرحلة ال
test هيعمله production

❖ اي حد محتاج اشرح اي جزئيه تاني سواء في ال **tasks** او حاجه من **topics** ال **sessinos** يقولي

عن أبي هريرة رضي الله عنه قال: سمعت رسول الله صلى الله عليه وسلم يقول: "ألا إن الدنيا
ملعونة ملعون ما فيها، إلا ذكر الله وما والاه، وعالم أو متعلم"