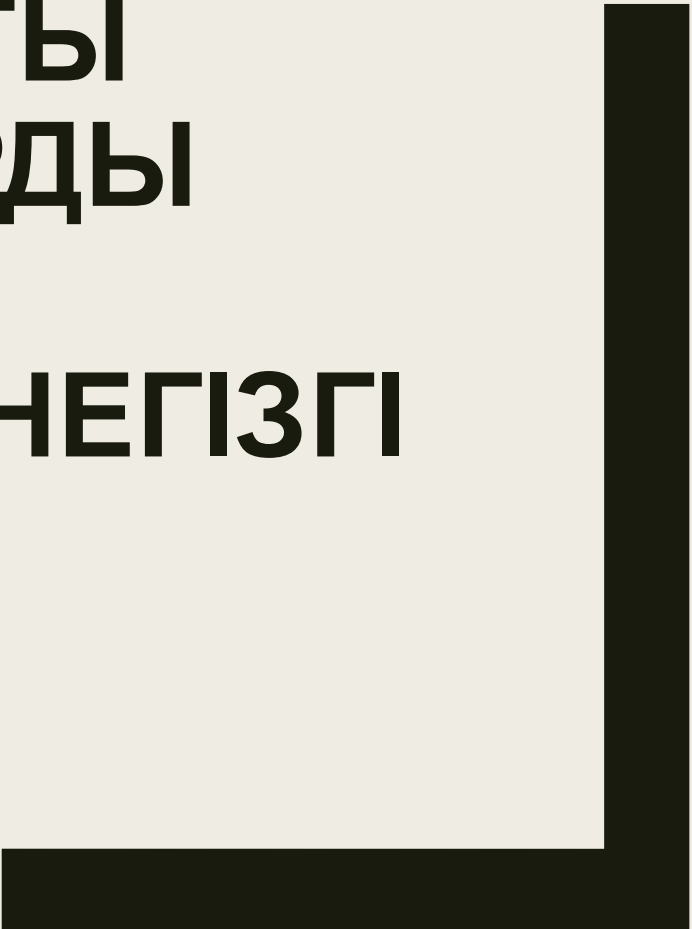


**ДӘРІС N°9. БҰЛТТЫ  
АРХИТЕКТУРАЛАРДЫ  
ЖОБАЛАУДЫҢ  
ЕРЕКШЕЛІКТЕРІ МЕН НЕГІЗГІ  
АСПЕКТІЛЕРІ.**



# Бұлтты дерекқорлар

- Бұлтты дерекқорлар - Amazon C2, GoGrid және Rackspace сияқты бұлтты есептеу платформаларында жұмыс істейтін дерекқорлар. Екі жалпы орналастыру үлгісі бар: пайдаланушылар бұлттық қызмет провайдері орналастыратын дерекқорларға кіру үшін қызметті тікелей сатып ала алады немесе виртуалды машина кескінін пайдаланып бұлтта дерекқорларды дербес іске қоса алады. Бұлтты дерекқорлардың арасында SQL-бағдарланған және NoSQL деректер үлгісін пайдаланатындар да бар.

# Қолдану үлгілері

- Бұлтта дерекқорды іске қосудың екі негізгі әдісі бар:
- Виртуалды машина кескіні - бұлттық платформалар дерекқорды іске қосу мүмкіндігі бар виртуалды машиналарды сатып алуға мүмкіндік береді. Пайдаланушылар өздерінің суреттерін орнатылған базамен жүктей алады немесе оңтайландырылған данасы орнатылған дайын суреттерді пайдалана алады. Мысалы, Oracle Amazon EC2 жүйесінде Oracle Database 11g Enterprise Edition виртуалды машинасы үшін дайын кескінді ұсынады.
- Деректер базасы қызмет ретінде - Кейбір бұлттық платформалар виртуалды машинаның қажеттілігін жоя алатын дерекқор қызметін ұсынады. Бұл жағдайда пайдаланушыға дерекқорды өзі орнату және жүргізу қажет емес. Оның орнына, қызмет провайдері дерекқорды орнату және қолдау жауапкершілігін алады. Мысалы, Amazon Web Services бұлттық қызметіне кіретін үш дерекқорды ұсынады: SimpleDB (NoSQL, мұнда деректер кілт-мән жұптарында сақталады), Amazon Relational Database Service (MySQL интерфейсі бар SQL-бағдарланған дерекқор) және DynamoDB.

# Архитектура және жалпы сипаттама

- Көптеген дерекқор провайдерлері пайдаланушылар дерекқор даналарын орнатып, конфигурациялай алатын веб-интерфейсті қамтамасыз етеді. Мысалы, Amazon Web Services веб-консолі дерекқор даналарын іске қосуға, суретті (сақтық көшірме сияқты) жасауға және статистиканы бақылауға мүмкіндік береді.
- Арнайы API қызметінің көмегімен негізгі дерекқорды басқаратын дерекқорды басқару компоненті ұсынылады. API пайдаланушыға ашық және оларға дерекқор даналарын қолдауға және масштабтауға мүмкіндік береді. Мысалы, Amazon Relational Database Service үшін API сізге дерекқор данасын жасауға, оның мазмұнын өзгертуге және лездік суреттер жасауға немесе бұрын жасалған суреттерден деректерді қалпына келтіруге мүмкіндік береді.
- Мұндай қызмет деректер қорын жүргізу үшін пайдаланылатын бағдарламалық жасақтаманың барлық стегін пайдаланушы үшін мөлдір етеді. Ол әдетте операциялық жүйені, деректер қорын басқару жүйесінің өзін және жұмыста қолданылатын үшінші тарап бағдарламалық жасақтамасын қамтиды. Қызмет провайдері осы бағдарламалық құралды орнатуға, түзетуге және басқаруға жауапкершілікті өз мойнына алады.
- Бұл қызмет дерекқордың ауқымдылығы мен қолжетімділігін қамтамасыз етеді. Сонымен қатар, масштабтау мүмкіндіктері әртүрлі жеткізушілер үшін ерекшеленеді - біреу оны автоматты түрде жасайды, ал басқалары пайдаланушыға API арқылы кеңейтуге мүмкіндік береді. Сондай-ақ, провайдерлер әдетте қызметтің жоғары қолжетімділігіне кепілдік береді (шамамен 99,9% немесе 99,99%).

# Деректер үлгілері

- Сондай-ақ реляциялық және реляциялық емес, NoSQL, мәліметтер қорын ажырату маңызды:
- SQL дерекқорлары - Nuodb, Oracle Database, Microsoft SQL Server және MySQL сияқты дерекқорлар. Олардың кез келгенін бұлтта іске қосуға болады және ол виртуалды машина кескіні немесе қызмет бола ма, ол тек провайдерге байланысты. SQL дерекқорларын масштабтау қиын, себебі олар бастапқыда бұлтқа арналмаған.
- NoSQL дерекқорлары Apache Cassandra, CouchDB және MongoDB сияқты дерекқорлар болып табылады. NoSQL дерекқорлары ауыр оқу/жазу жүктемесіне төтеп беруге, сондай-ақ өсіруге және кішірейтуге оңай жасалған және олар бастапқыда бұлттық платформалар үшін жасалған. Дегенмен, қазіргі заманғы бағдарламалардың көпшілігі SQL көмегімен жасалған, сондықтан NoSQL деректер базасымен жұмыс істеу көбінесе қолданбалы кодты толығымен қайта жазуды талап етеді.
- Осылайша, әрбір пайдаланушы тобы үшін қосымшаны бұлтта орналастырған жағдайда ресурстардың бекітілген көлемі емес, тек қажеттісі ғана бөлінеді, яғни. «Шығын» «бұлттық» ресурстарды пайдалану негізінде алынады.

# Көп пәтерлі архитектураға көшу кезеңдері:

- Арнайы архитектура. Әрбір пайдаланушы тобы қолданбаның (қызметтің) жеке көшірмесін пайдаланады. Әрбір көшірме тек өзінің пайдаланушылар тобына бағытталған, осылайша әрбір көшірмеде пайдаланушы параметрлерін сақтайтын жеке кеңейтім бар. Бұл модельдің барлық тиімсіздігі пайдаланушылар санының өсуімен анық болады. Сондай-ақ масштабтау арқылы тиімділікті арттыру мүмкін емес, өйткені пайдаланушылар қолданбаның әртүрлі даналарын пайдаланады.
- Арнайы архитектура. Бұл жағдайда қолданба немесе қызмет конфигурация арқылы әрбір нақты пайдаланушы үшін конфигурацияланады. Анау. әр пайдаланушы көшірмелері бірдей болғанына қарамастан, қосымшаның жеке көшірмесімен жұмыс істейді. Есептеу қуаты қолданбаның әртүрлі даналары арасында бөлісілмейді, бұл сонымен қатар тиімділік артуының масштабталуына жол бермейді. Қолданбаны бөлудің өзі виртуалды және физикалық болуы мүмкін.
- Көп пәтерлі архитектура. Пайдаланушыға арналған конфигурация тек өзін-өзі конфигурациялау құралдарын пайдаланып конфигурациялау арқылы жасалады. Сонымен қатар көлденең масштабтау мүмкіндігі жоқ, яғни. өнімділікке тек тік масштабтау арқылы қол жеткізуге болады.
- Масштабталатын архитектура. Бұл модель көп жалға алушылық тәсілді, сондай-ақ көлденең масштабтау мүмкіндігін қолдайды. Бұл ортақ пулға жаңа қолданба даналарын қосады. Жалпы жүктеме бүкіл инфрақұрылымға бөлінеді. Архитектура конфигурация арқылы конфигурацияланады.