# Middle DevOps - Linux

## 1. Опишіть архітектуру ядра Linux.

Ядро Linux має модульну архітектуру, яка дозволяє додавати та видаляти функціональність у вигляді модулів. Основні компоненти архітектури ядра Linux включають:

*Системні виклики (System Calls):*

Ядро забезпечує механізми для взаємодії зі зовнішніми програмами через системні виклики. Це дозволяє програмам звертатися до ресурсів системи, таких як файлова система, мережеві можливості та інші.

*Менеджер пам'яті (Memory Management):*

Ядро відповідає за управління фізичною та віртуальною пам'яттю комп'ютера. Це включає управління відображенням віртуальної пам'яті на фізичну, роботу зі сторінками пам'яті та інші аспекти.

*Планування процесів (Process Scheduling):*

Ядро відповідає за планування виконання процесів на процесорі. Це включає вибір процесів для виконання, розподіл часу процесора та призначення пріоритетів.

*Мережеві протоколи (Networking Protocols):*

Ядро містить реалізації різних мережевих протоколів, які дозволяють обмін даними через мережу. Це включає підтримку протоколів TCP/IP, UDP, ICMP та інших.

*Файлові системи (File Systems):*

Ядро надає підтримку різних файлових систем, таких як ext4, btrfs, NTFS та інші. Це дозволяє працювати з файлами та каталогами на зберігаючих пристроях.

*Процеси та потоки (Processes and Threads):*

Ядро керує процесами та потоками в операційній системі, забезпечуючи їх виконання та взаємодію.

## 2. Що таке ядро операційної системи і яке його призначення?

Ядро операційної системи (ОС) є центральною частиною операційної системи, яка відповідає за управління ресурсами комп'ютера та надання середовища для виконання програм.

Призначення ядра операційної системи включає:

*Управління ресурсами:* Ядро контролює доступ до фізичної пам'яті, процесора, введення-виведення та інших ресурсів комп'ютера. Воно розподіляє ресурси між різними програмами та процесами, забезпечуючи їхню взаємодію та виконання.

*Взаємодія з апаратним забезпеченням*: Ядро взаємодіє з апаратним забезпеченням комп'ютера, керуючи взаємодією з пристроями введення-виведення, дисками, мережею та іншими пристроями.

*Забезпечення системних викликів*: Ядро надає інтерфейс для взаємодії програм з операційною системою через системні виклики. Це дозволяє програмам звертатися до різних функцій ОС, таких як робота з файлами, мережею, пам'яттю та іншими ресурсами.

*Управління процесами та потоками*: Ядро відповідає за планування виконання процесів та управління потоками в операційній системі.

*Підтримка мережі та файлових систем*: Ядро надає підтримку мережі та різних файлових систем для забезпечення взаємодії з іншими комп'ютерами та зберігання даних.

## 3. Опишіть загальні частини файлової системи Unix/Linux, архітектуру файлової системи.

Загальні частини файлової системи Unix/Linux:

*Кореневий каталог (Root Directory):*

Кореневий каталог позначений символом "/", він є стартовою точкою для всієї файлової системи. Всі інші файли та каталоги розташовані в межах цього каталогу.

*Каталоги (Directories):*

Файлова система Unix/Linux містить каталоги, які дозволяють організувати файли у структуровану ієрархію. Кожен каталог може містити інші файли та каталоги.

*Файли (Files):*

Файли представляють дані, які можуть бути текстовими, виконуваними програмами, зображеннями тощо. Файли можуть бути організовані у каталогах для зручності управління та доступу.

*Символічні посилання (Symbolic Links):*

Unix/Linux дозволяє створювати символічні посилання, які є посиланнями на інші файли або каталоги. Вони надають зручний спосіб посилання на різні ресурси в межах файлової системи.

*Спеціальні файли (Special Files):*

Файлова система містить спеціальні файли, такі як пристрої (наприклад, /dev/null для відкидання даних) та сокети, які використовуються для міжпроцесного зв'язку.

**Архітектура файлової системи Unix/Linux:**

Файлова система Unix/Linux має ієрархічну структуру, що починається з кореневого каталогу "/" і включає в себе різні каталоги та файли. Основні аспекти архітектури файлової системи включають:

*Кореневий каталог* ("/"): Це початковий пункт в ієрархії файлової системи*.*

*Каталоги та підкаталоги*: Файлова система містить каталоги, які можуть містити інші каталоги та файли.

*Файли*: Дані в файловій системі представлені у вигляді файлів, які можуть мати різний тип та призначення.

*Символічні посилання*: Unix/Linux дозволяє створювати символічні посилання на файли та каталоги для зручності управління ресурсами.

*Спеціальні файли*: Файлова система містить спеціальні файли, такі як пристрої та сокети, які надають доступ до апаратного забезпечення та мережевих можливостей.

## 4. У чому різниця між RedHat і Debian?

## 5. У чому різниця між /proc and /sys?

## 6. Ситуація: показує, що на диску зайнято 50% місця, а створити файл навіть під root користувачем не можемо. У чому проблема?

## 7a. Що може створювати високе навантаження на CPU (процеси застосунків споживають дуже мало ресурсів CPU)?

*Мультизадачність*: Якщо ви виконуєте багато програм одночасно, то це може призвести до великого навантаження на CPU, навіть якщо окремі процеси застосунків не споживають багато ресурсів.

*Фонові процеси*: Деякі застосунки можуть запускати фонові процеси, які використовують ресурси CPU, навіть якщо основні процеси застосунку не активні.

*Погано оптимізований код*: Іноді програми можуть мати погано оптимізований код, який призводить до великого навантаження на CPU при виконанні навіть малих завдань.

*Віруси та шкідливі програми*: Шкідливі програми можуть використовувати ресурси CPU без вашого дозволу, що може призвести до високого навантаження.

*Оновлення та фонові завдання ОС*: Операційні системи можуть виконувати фонові завдання, такі як оновлення або планові завдання обслуговування, які можуть спричиняти велике навантаження на CPU.

Якщо ви стикаєтеся з високим навантаженням на CPU, вам може бути корисно використовувати системні засоби моніторингу, щоб виявити, які процеси споживають найбільше ресурсів, а також виконати антивірусну перевірку для виявлення можливих шкідливих програм.

## 7b. Ми видалили файл, який відкрив застосунок. Як нам його відновити?

## 8. Як знайти PID процесу, його стартові параметри?

## 9. Як перевірити, чи відкритий порт на віддаленому хості, локальному хості?

10. Як шукати файл за його вмістом?

11. Що таке SSH, як організувати доступ на сервер без пароля або з певних хостів? Як обмежити доступні для виконання команди?

12. Як перевірити спожиті ресурси під час сеансу SSH?

13. Що означає дозвіл на файл 755?

14. Що таке SELinux і навіщо він потрібен?

15. Як визначити PCI-пристрій у системі, наприклад RAID controller?

16. Як перейменувати пристрій, наприклад мережеву карту чи диск?

17. Що таке LVM? Які знаєте приклади використання?

18. Що таке root reserved space?

19. Що таке exit code та як його дізнатися?

20. Чому вивід df -h показує, що на диску зайнято мало місця, але система не дає записати файл із повідомленням no space left on device?

21. У чому різниця між command1 & command2 та command1 && command2, а також command1 && command2 || command3?

22. З мережі різко зріс вихідний трафік на 25-й порт. Як, маючи доступ на гейтвей, виявити шкідника з внутрішньої мережі?

23. Як затюнити параметри Linux Kernel?

24. Що таке ulimits?

25. У чому різниця між символічними та hard links?

26. Що таке фрагментація ext3 та ext4?

27. Навіщо файлові системи ext\* резервують 5% місця?

28. Як збільшити розмір файлової системи?

29. Чи можемо ми зменшити розмір файлової системи?

30. Що таке chroot і навіщо він потрібний?

31. У нас є Linux box з 2 Гб оперативної пам’яті та Java-застосунок, який намагається виділити 4 Гб під час запуску. Чи вдасться це?

32. Є програма, яка читає файл, який користувач намагається видалити. Що трапиться? Чи можна видалити цей файл? Чи можна відновити цей файл?

33. Які механізми створення процесів у Linux ви знаєте?

34. Порівняйте systemd та init system.

35. У вас є папка з великою кількістю файлів, і ви хочете видалити всі файли з іменами, що починаються на A (велика літера). Але команда rm -f A\* видає Argument list too long. Як можна видалити ці файли?

36. Ви починаєте видаляти файли першим методом з попереднього питання, але кожен rm запитує підтвердження. Це дуже довго. Як можна прискорити цю операцію?

# Mddle DevOps - Networks

## 37. Розкажіть про модель OSI. Опишіть функції та призначення кожного рівня.

## 38.Які мережеві топології ви знаєте? Опишіть різницю між ними.

39.Навіщо потрібна IP-адреса, якщо MAC-адреса унікальна? Хіба ми не можемо спілкуватися тільки за MAC-адресою?

40.У чому різниця між концентратором і комутатором L2 у мережах Ethernet?

41.Що таке Vlan і навіщо існує поділ на віртуальні локальні мережі?

42.Який номер порту використовують для PING-комунікації?

43.Що таке сеанс зв’язку? Який алгоритм використовує TCP для доставки?

44.У чому основна відмінність між TCP і UDP?

45.Навіщо нам маршрутизатор за замовчуванням?

46.Як хост вирішує DNS за замовчуванням?

47.Комп’ютер почав отримувати IP-адресу з іншої мережі (є підозра, що в мережі працює інший DHCP-сервер): як його знайти і від’єднати? Які є методи захисту від такої проблеми?

48.Ми будемо мігрувати сайт на нову IP-адресу. Як зробити, щоб користувачі цього практично не помітили?

49.Що таке socket?

50.Як дізнатися, які віддалені хости під’єднуються до хосту через порт 8888? (за допомогою команд і не використовуючи /proc чи /sys).

51.Ми маємо кілька мережевих карт. Як збільшити пропускну спроможність сервера?

52.Як перевірити відкриті порти на віддаленому сервері без команди Netcat чи Nmap Linux?

Container orchestration

53.У чому переваги Kubernetes як платформи?

54.Що таке control plane та з яких компонентів складається?

55.Які CNI ви використовували та чим вони відрізняються?

56.Чим відрізняється managed Kubernetes від self-deployed?

57.Яким чином можна контролювати розміщення подів у кластері? (taints/tolerations, affinities, topologies etc)

58.Скейлінг кластеру. Cluster autoscaler vs HPA vs VPA? Як зробити zero-downtime node decommission/cluster upgrade? PDB? Lifecycle hooks?

59.Які існують способи для зовнішнього доступу до кластеру? ingress, node port, port-forward тощо.

60.З яким PID запускається процес у контейнері?

61.Що краще використовувати для ізоляції оточення — Vagrant чи Docker?

62.Який інструмент оркестрування контейнерів використовували? (Swarm, Kubernetes, Openshift, Rancher тощо.)

63.Що відбувається в Kubernetes після запуску kubectl (API, ReplicaSet Controller, storage back-end, scheduler, kubelet, worker node, pod)?

64.Яка різниця між pod та контейнером у K8s?

65.Як ми можемо зробити будь-який мікросервіс, який працює на K8s, доступним із зовнішнього середовища?

Віртуалізація і контейнеризація

66.Які типи віртуалізації ви знаєте?

67.Як працює Docker на macOS/Windows?

68.Що таке Docker-image і Docker-контейнер? Як вони між собою пов’язані?

69.Які основні відмінності між контейнерами докерів і віртуальними машинами?

70.Що таке image layer? Яка максимальна кількість layers можлива? Чому треба намагатися мати малу кількість layers? Яка оптимальна кількість?

71.Як у віртуальній машині змінити розмір диска після створення? Що треба зробити з гостьовою ОС?

72.Як у Docker реалізовано обмеження ресурсів?

73.Існує віртуальна машина, до якої втрачено доступ. Як, маючи доступ до її диска, відновити root пароль/SSH-ключ?

74.Оптимізувати Dockerfile, пояснити, що і чому так:

FROM golang

RUN apt install -y pkg1 pkg2 pkgN # Dependencies for app

COPY. .

RUN go build -o app main.go

CMD ./app

75.Що таке IPVS та який функціонал у нього є?

76.Яка структура API у Kubernetes?

77.Що таке operators і для чого вони потрібні?

CI/CD

78.Які стадії мають бути в будь-якому пайплайні (lint, test, build, deploy etc)?

79.Як і де зберігати build artifacts?

80.Що таке артефакт?

81.Є два бренчі: dev і stage. Ми закинули Dockerfile у dev, а потім збілдили у dev і stage. Це буде одним артефактом чи різними?

82.Що ви використовували для автоматизації налаштування Jenkins and GitLab CI?

83.Порівняйте CI інструментів: Jenkins, GitLab CI, AWS Code Pipeline, GCP cloudbuild, GitHub actions, Circle CI.

84.Deployment strategies. Які існують і чим відрізняються (recreate, blue-green, canary etc)?

85.Яким чином реалізувати СI/CD для програми, яка залежить від кількох інших програм?

86.GitOps. В чому його переваги та недоліки?

Clouds and Automation

87.Яка роль і переваги хмарних сервісів для DevOps?

88.Що таке immutable infrastructure? Як досягти? У чому переваги та недоліки? Packer, AMI тощо.

89.Структура Terraform. Як організувати multi-environment project? Terraform workspaces?

90.Найкращі практики з використання багатьох Terraform states.

91.Як організувати доступ розробницькій команді до AWS/GCP/Azure? Role-based access, assume role, SSO.

92.Що таке Terraform provider, module?

93.Як версіонувати Terraform modules?

94.Коли потрібно використовувати local-exec та remote-exec?

95.Що таке golden image та як його створити?

Monitoring/Logging

96.Як моніторинг допомагає підтримувати всю архітектуру системи?

97.Які інструменти моніторингу ви використовували?

98.Що таке медіана та процентиль?

99.Що таке SLI, SLO, SLA? Навіщо це потрібно?

100.Архітектура системи для збору логів, ELK, EFK etc. Як зберегти логи у разі відмови сховища? Чи потрібно використовувати для цього брокер повідомлень? Чи потрібно робити throttling/rate limits?

101.Prometheus long-term storage. Які варіанти?

102.Як працює Prometheus?

103.У чому принципова відмінність між Grafana і Kibana?

104.В чому головна відмінність між Ansible and Terraform?

105.Що таке SAAS monitoring та які види знаєте?

106.Якщо ви використовуєте Datadog/NewRelic, то як нам відстежувати падіння інструментів моніторингу?

107.Що таке distributed tracing та error tracking systems? Як ви думаєте, коли варто їх використовувати?

Information Security

108.У чому різниця між RBAC та ABAC?

109.У чому полягає XSS атака? SQL injection? Що таке CSP?

110.Які базові заходи можна вжити для захисту SSH-з’єднання?

111.Root-пароль невідомий чи загублений. Яка процедура відновлення?

112.Як керувати правами на файловій системі в Linux?

113.Що таке Firewall?

114.Чим відрізняється stateless від stateful фаєрволів?

115.Скільки таблиць у iptables?

116.Чи можна налаштувати трансляцію NAT за допомогою iptables? Яку таблицю варто використати?

117.Яку таблицю використовують для зміни заголовків пакетів?

118.Якщо вам ламають Linux-сервер, то як більш ефективно блокувати трафік з IP-адрес?

119.Принцип роботи GCP Firewall: чи можемо ми профільтрувати трафік на Load Balancer?

120.Що таке SELinux?

121.Чи можна повністю від’єднати SELinux на льоту?

122.З якими secrets management systems ви працювали?

123.У нас є сервер NAT, і ми хочемо забезпечити доступ за ІР до сервера зовні. Як нам це реалізувати?

123.Щоб потрапити на сервер клієнта, треба залогінитись на 4+ jump хоста. Як це автоматизувати? Де ми зберігатимемо наш SSH-ключ?

Development

125.Що таке cookies? Для чого потрібні? JWT?

126.Що таке feature toggles та навіщо вони?

127.Що таке TDD (Test Driven Development) та BDD (Behaviour Driven Development)?

Databases

128.Що таке індекс і що таке ключ?

129.Які переваги та недоліки індексів?

130.Уявіть, що ви розробляєте систему білінгу, яка має обробляти тисячі рахунків. Яку стратегію оновлення даних ви б обрали?

131.Які методи найчастіше використовують для масштабування реляційних баз даних?

132.Опишіть механізм транзакцій БД.

133.Як ми можемо видалити таблицю чи базу даних?

134.Як знайти повільні запити у MySQL/PostgreSQL?

135.Які SQL-оператори маніпулювання даними ви знаєте?

136.Чи можна вивести список баз даних/таблиць через CLI? Як ми можемо перемикатися між базами даних MySQL/PostgreSQL?

137.Які storage engines в MySQL ви знаєте? Які відмінності?

138.Як реалізовано реплікацію MySQL master-master? Скільки серверів MySQL може бути залучено в такій взаємодії?

139.Як працює реплікація MySQL/PostgreSQL? Які параметри мають бути налаштовані для реплікації?

140.Порівняйте SQL і NoSQL.

141.Sharding vs replication?

142.Які є види індексів? Коли і для чого використовувати?

143.Вимоги до схеми БД. Character sets, collations, default, not null тощо.

144.Ми мігруємо MySQL/PostgreSQL з on-prem у хмару. Як нам зробити це з мінімальним даунтаймом?

145.Навіщо та як тестувати перформанс баз даних?

Практичні завдання

146.Напишіть Terraform module для інфраструктури тестового сервісу у AWS.

147.Напишіть hello-world програму мовою на ваш вибір і сформуйте для неї helm chart/kustomize.

148.Як організувати деплой без downtime?

149.Опишіть способи troubleshooting для Docker-контейнера.

150.Розібрати і пояснити структуру CI/CD pipeline (на прикладі gitlab.yml).

151.Продемонструйте навички роботи з GitOps, опишіть деплоймент простенької програми.

152.Як організувати деплой вебзастосунку, запущений на кількох серверах без (або з мінімальним) downtime?

153.Як за допомогою Ansible дізнатися default gateway для пулу серверів, і якщо він відрізняється від бажаного, записати рядок «hostname: gateway» у файл на локальній машині?