

Загальна кількість слайдів – 10-15 (виходити з того, що Ви будете обмежені за часом – максимум 12 хвилин (доповідь + демонстрація) + 10 хвилин відповіді на запитання)

Основні складові:

1. Тема – назва проекту

2. Мотивація – чому вирішення завдання, яке Ви обрали – це важливо (хай навіть особисто для Вас)

3. Актуальність – це об'єктивне підтвердження важливості вирішення обраної задачі, оскільки багато людей вирішували їх у недавньому минулому і вирішують її зараз у різний спосіб.

4. Мета – забезпечити, збільшити, скоротити, зробити безпечнішим, прискорити тощо. і т.п. у контексті вирішення Вашого завдання.

5. Завдання – розбиття Вашого спільного завдання на дрібніші приватні (які легше вирішити).

6. Проектування (обґрунтування, чому) (Вибір стеку технологій, які будуть використовуватися при реалізації проекту)

7. Реалізація (якщо є тестування – відмінно) (Послідовність реалізації проекту, виклики, досягнення, основні результати)

8. Демонстрація (показати, як у реалі працює Ваш проект)

9. Висновки (Що вийшло, що хотілося б покращити і зразкові шляхи для покращення)

10. До питання про мінімальну/максимальну функціональність проекту (фінального тягу).

Мінімум:

- хост з гітом (Linux, MacOS, Windows: [можливо на віртуалці]);

- В якості смислової частини можна вибрати html сторінку (index.html) з вмістом на Ваш розсуд.

- хмарний репозиторій (GitHub, GitLab, BitBucket), куди Ви робитимете commit/push;

- CI/CD framework (Jenkins) він може бути з "білим" ip (від провайдера або, що краще, в AWS) або "сірим" (за домашнім роутером у якого теж "сірий" ip [NAT]). У разі сірого не можете використовувати (без додаткових коштів) GitHubWebHook, тобто. використовувати інші варіанти з лекції. Тобто. основне призначення дженкіса - стягнути оновлений проект з гітхабу і зібрати його (build), а потім помістити в сховище артефактів (делівері) або розгорнути на таргет сервері (деплой);

- Configuration management tool (Ansible) – можна використовувати для підняття машинок та початкового настроювання оточення Вашого проекту;

- Необхідно, щоб на дженкінс майстра відбувалося перенаправлення джобів на його ноди (слейви). У випадку з веб-сторінкою, цей пункт можна опустити, відразу деплоїти на таргет сервер.

-Докер і контейнеризацію можна використовувати наступним чином - організувати запуск дженкінса і його нод в контейнерах і таргет-сервера також в контейнері.

Все, що ви зробите, у будь-якому випадку добре. Особливо тим, хто у темі ДевОпс дуже короткий час.

При демонстрації працездатності проекту внести мінімальні зміни до коду проекту, зробити коміт, пуш для оновлення клауд-репозиторію (GitHub, GitLab, BitBucket) та відпрацювання дженкінсом змін.

Бажано, щоб новий білд за проектом мав візуальне відображення.

По середнім/максималкам:

Середній + рівень передбачає реалізацію випускного завдання відповідно до вищепереліченого, проте як основа проекту може бути обраний GitHub-прототип середнього/великого проекту на Java (наприклад, PetClinic) із застосуванням maven (ant, gradle), тестуванням та розгортанням. Також можливий варіант Python проекту на основі існуючого репозиторію GitHub.

Максималка – унікальний (авторський) проект (Java, Python, .Net, etc.) з використанням елементів IaC (Terraform, CloudFormation, etc), залучення хмарних сервісів СУБД, застосування технологій розгортання, наприклад, blue-green deployment.