✅1. Mock Answer: Tell Me About Yourself

"Hi, I'm William, a fresh software engineer with hands-on experience in both frontend and backend development. I've mainly worked with technologies like Vue.js, TypeScript, Node.js, and Python, and I enjoy solving real-world problems through code.

While my direct experience with Linux engineering is still developing, I've used Linux-based environments for coding, running servers, Docker containers, and general deployment tasks. I'm now actively looking to deepen my expertise in Linux system administration, especially in an infrastructure-focused, globally impactful environment like Hilti. I'm excited about this opportunity to bridge my software skills with system-level engineering and contribute meaningfully to the team."

✅2. Common Linux Concepts & Commands (Cheat Sheet)

File System Hierarchy:

- `/etc`: configuration files
- `/var/log`: system logs
- `/home`: user directories
- `/usr/bin`: user command binaries

Important Commands:

```
ls        # list directory contents
cd        # change directory
cp, mv    # copy, move files
rm        # delete files
chmod     # change file permissions
chown     # change file ownership
ps, top   # view running processes
systemctl # manage services (start, stop, enable)
journalctl # view system logs
apt, yum  # package managers (Ubuntu, RHEL)
scp       # copy files remotely
ssh user@ip # connect to remote Linux server
```

Useful Concepts:

- SSH = secure remote access
- Services = daemons (e.g., nginx, sshd)
- Log files = `/var/log/syslog`, `/var/log/messages`
- Package managers: `apt` (Debian/Ubuntu), `yum/dnf` (RHEL/CentOS)

✅3. CI/CD & Infrastructure as Code (IaC)

CI/CD:

- CI (Continuous Integration): Automatically build/test your code when you push.
- CD (Continuous Deployment): Automatically deploy changes to servers.
- GitLab CI/CD: Use `.gitlab-ci.yml` to define build/deploy stages.

IaC Concepts:

- Infrastructure as Code = manage servers/config using code/scripts (not manual work).
- Tools: Puppet, Ansible, Terraform.
- Puppet: declarative language to define how servers should be configured.

✅4. Quick Cram Guide: Key Tools for This Role

🔸 **Puppet**

- Configuration management tool
- Declarative: define *what* the system should look like, Puppet makes it so.
- Common uses: manage users, install packages, control services.
- Example: ensure nginx is always installed and running.

🔸 **Red Hat Satellite**

- Tool for managing RHEL systems at scale.
- Helps automate patching, provisioning, and configuration.
- Often works with Puppet for system-wide management.

🔸 **GitLab CI/CD**

- Automation pipelines for build, test, deploy.
- You define stages in `.gitlab-ci.yml`.
- Example:

```
stages:
  - build
  - deploy

deploy-prod:
  stage: deploy
  script:
    - echo 'Deploying to production...'
```

🔸 **Microsoft Azure (basic knowledge)**

- Cloud platform similar to AWS/GCP.
- You might work with:
- Azure VMs (Linux servers on cloud)
- Azure DevOps pipelines
- Azure Resource Manager (for IaC)

🔸 **Linux Security Terms**

- **CVE**: Common Vulnerabilities and Exposures. Public list of known security flaws.
- **SCAP**: Security Content Automation Protocol. Used to check system compliance.
- Understand: applying updates, checking for vulnerabilities, securing SSH.

✅5. How to Be Honest but Positive

"I've mainly used Linux from a developer's perspective — setting up environments, deploying apps, debugging logs. I haven't yet worked in a dedicated sysadmin or platform engineering role, but I'm eager to learn tools like Puppet, Red Hat, and dive deeper into cloud and security topics."

"I'm confident that my background in engineering and scripting, plus my interest in system-level work, gives me a strong foundation to grow into this role."

✅6. Questions You Can Ask Them

1. "What does a typical onboarding process look like for someone new to Linux engineering but experienced in development?"
2. "How closely do the application and platform teams collaborate at Hilti?"
3. "Are there opportunities to cross-skill across infrastructure and DevOps/cloud areas?"
4. "How does the team stay up-to-date with new Linux security standards or automation tools?"

Let me know if you'd like me to simulate mock interview questions next or dive deeper into any specific tool!