Explain AWS CodeBuild features and how you are using in your project ?

When I started working with **AWS CodeBuild**, the first thing that impressed me was how much setup it took *off my plate*. Here's how I break it down based on real usage:

🔧 **Managed Build Environments**
"One of the biggest time-savers for me was the pre-configured environments. I didn't have to worry about setting up Linux, Windows, or macOS systems. It came with runtimes like Python, Java, Node.js, .NET Core, and even Android. Build tools like Maven and npm were ready to go. It just worked out of the box."

🛠️ **Custom Build Environments**
"But for specialized cases, I had the flexibility to bring my own Docker image. I just pushed it to ECR, and CodeBuild used it exactly how I wanted—especially useful when working with the full .NET Framework."

⚙️ **Flexible Compute Types**
"Another thing that mattered: choosing between EC2-based compute or Lambda-based compute. EC2 gave me more control, but Lambda was blazing fast to start and scaled automatically. I used it to reduce build queue delays during peak hours."

🔗 **Source Code Integrations**
"Integrating with source repos was seamless—whether it was GitHub, Bitbucket, CodeCommit, or even S3. I enabled webhooks, and builds got triggered right on commits. It kept my CI truly continuous."

🔁 **CI/CD Pipeline Integration**
"CodeBuild plugged into CodePipeline like Lego blocks. From code commit to deploy—it was automated. I even integrated Jenkins once for a hybrid workflow with our legacy builds."

🧪 **Parallel Test Execution**

"A game-changer was parallel test execution. I split our tests across multiple environments, and build times dropped significantly. CI feedback was faster—developers loved it."

🐳 Persistent Docker Server

"CodeBuild introduced a dedicated Docker server per project. That helped us cache Docker layers and reduced build times for containerized apps. We saw almost 30% improvement on large builds."

💾 Build Caching

"I configured both local and S3 caching for dependencies like Maven repositories and npm modules. It prevented redundant work and made repeat builds much faster."

🔐 Security & Permissions

"Security is tight. Artifacts are encrypted using KMS keys. Each project uses an IAM role, so I can control access down to the API level. Also, every build runs in an isolated container—no leakage or contamination."

📊 Monitoring & Notifications

"Monitoring was simple. I tracked build status, commit ID, and duration through the console and CloudWatch. Logs were streamed automatically. Plus, I set up SNS notifications for build failures to alert the team on Slack."

💸 Cost Efficiency

"It's truly pay-as-you-go. We only pay for the minutes used. And yes, the free tier covers a lot of ground for side projects or early testing. We optimized costs further with reserved capacity fleets for high-frequency builds."

✅ Why I Stick With CodeBuild

🧰 **No server management:** I don't maintain or patch anything—AWS does it.
📈 **Scales automatically:** Whether 1 job or 100, it handles them.
⚡ **Fast and efficient:** Caching, parallelism, and optimized compute save tons of time.
💰 **Cost-effective:** Simple billing—only pay for what we use.
🔗 **Well-integrated:** Part of AWS Code suite—CodeCommit, CodePipeline, CodeDeploy.
🔐 **Secure:** KMS, IAM, isolated builds—what more can I ask for?
🚀 **Agility:** Quick feedback cycles → faster releases → happier teams.

🧱 Advanced Configuration Experience (Build Project Settings)

When I configure a project via **console or CLI**, here's how I typically set it up:

1. **Project Name & Description:** Clearly define the build project.
2. **Source:** Usually GitHub or CodeCommit. I set up webhooks for instant triggers.
3. **Primary Source Version:** I can specify a branch, tag, or even a specific commit.
4. **Environment:**
   - Managed or custom image
   - Compute type: EC2, Lambda, or reserved fleet

        - IAM service role for permissions
5. **Buildspec:** I prefer keeping buildspec.yml in the repo, but inline works for quick tests.
6. **Artifacts:** Pushed to S3 or passed to CodePipeline, encrypted via KMS.
7. **Logs:** Always enabled CloudWatch logs. Sometimes added S3 for archives.
8. **VPC Configuration:** Used when builds needed to access internal databases.
9. **Environment Variables:**
    - Used PLAINTEXT for local testing
    - Pulled secure values from **SSM Parameter Store** or **Secrets Manager**

🧾 **buildspec.yml - Real Power in Simplicity**

My buildspec files typically follow this format:

yaml

```
------
version: 0.2

env:
  parameter-store:
    DB_URL: "/my-app/prod/database/url"
  secrets-manager:
    API_KEY: "my-app/prod/api_key"

phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm run build
  post_build:
    commands:
      - aws s3 cp ./build s3://my-bucket/

artifacts:
  files:
    - '**/*'
  base-directory: build
```

"I use parameter-store for DB URLs and secrets-manager for sensitive credentials. It’s secure and manageable."

This is how AWS CodeBuild fits into my day-to-day DevOps workflow—it gives me **speed**, **security**, and **automation** without the usual server headaches.

What do you need to do to use a custom build environment in AWS CodeBuild?

A. Install the custom tools directly in CodeBuild console

B. Package the tools in a Docker image and upload to ECR or Docker Hub

C. Modify the EC2 instance type
D. Use Lambda instead of EC2

✅ **Correct Answer:** B. Package the tools in a Docker image and upload to ECR or Docker Hub

---

Which of the following is NOT a supported source provider for AWS CodeBuild?
A. GitHub
B. CodeCommit
C. Bitbucket
D. Google Drive

✅ **Correct Answer:** D. Google Drive

Which caching options are available in AWS CodeBuild?
A. Only Amazon S3
B. Only local disk
C. Amazon S3 and Local cache
D. Amazon EFS only

✅ **Correct Answer:** C. Amazon S3 and Local cache

Which method is used in CodeBuild to fetch environment variables stored securely in AWS?
A. PLAINTEXT only
B. YAML file
C. SSM Parameter Store and Secrets Manager
D. AWS CloudTrail

✅ **Correct Answer:** C. SSM Parameter Store and Secrets Manager

---