1. **Fabric**:
   * Fabric is a high-level Python library designed to execute shell commands remotely over SSH, making it easy to automate deployments and system administration tasks.
2. **Ansible**:
   * Although Ansible itself is a tool written in Python, it can be extended and customized with Python. Ansible modules can be written in Python to automate and manage IT infrastructure.
3. **Paramiko**:
   * Paramiko is a Python implementation of the SSHv2 protocol, providing both client and server functionality. It's useful for creating secure connections to remote systems for executing commands and transferring files.
4. **Boto3**:
   * Boto3 is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python, allowing you to write scripts and applications that use AWS services like S3, EC2, DynamoDB, and more.
5. **Docker SDK for Python**:
   * This SDK allows you to interact with the Docker daemon through Python, enabling you to manage Docker containers, images, networks, and volumes programmatically.
6. **PyYAML**:
   * PyYAML is a YAML parser and emitter for Python, useful for working with configuration files commonly used in DevOps tools like Ansible, Kubernetes, and Docker Compose.
7. **Requests**:
   * Requests is a simple and elegant HTTP library for Python, ideal for interacting with RESTful APIs and web services.
8. **JenkinsAPI**:
   * JenkinsAPI lets you interact with Jenkins from within Python scripts, providing access to the Jenkins REST API for managing jobs, builds, and other resources.
9. **Kubernetes (K8s) Python Client**:
   * This client library allows you to interact with Kubernetes clusters using Python, enabling automation of deployment, scaling, and management of containerized applications.
10. **PyTest**:
    * PyTest is a testing framework that can be used for writing simple unit tests as well as complex functional testing for applications, which is crucial in a DevOps pipeline.
11. **SaltStack**:
    * SaltStack provides remote execution capabilities for any infrastructure management and configuration tasks. It can be used programmatically with Python to manage server states and configurations.
12. **Celery**:
    * Celery is an asynchronous task queue/job queue that is based on distributed message passing. It's useful for handling asynchronous tasks in a DevOps environment, such as background job processing.
13. **Supervisor**:
    * Supervisor is a system for controlling and monitoring processes. It can be used to manage background processes, and it has a Python API for programmatic control.