DevOps Shack by Aditya Jaiswal

10 Real-Time Python Automation Scripts

Script 1 | Email Campaign

Single Recipient

```
import smtplib
from email.mime.text import MIMEText

def send_email(subject, body, to_email):
    # Email settings
    sender_email = 'jaiswaladi246@gmail.com'
    sender_password = 'your_password_here'  # Use App Password if 2-Step Verification is enabled

    # Create email message
    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From'] = sender_email
    msg['To'] = to_email

    # Sending email
    with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
        server.login(sender_email, sender_password)
        server.sendmail(sender_email, to_email, msg.as_string())

# Example usage
send_email('Automation Test', 'This is an automated message.', 'recipient@example.com')
```

Multiple Recipients

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import os

def send_individual_emails(subject, body, recipients, pdf_path=None):
    # Email settings
    sender_email = 'jaiswaladi246@gmail.com'
    sender_password = 'your_password_here'  # Use App Password if 2-Step Verification is enabled

    # Create base HTML body template for beautification
    html_body_template = """<html>
    <body style="font-family: 'Segoe UI'; background-color:#f4f4f4; padding:20px; color:#333;">
        <div style="max-width:600px; margin:0 auto; background-color:white; padding:20px; border-radius:10px; box-shadow:0 0 10px rgba(0,0,0,0.1);">
            <h2 style="color:#4CAF50; text-align:center;">Welcome to Batch-7!</h2>
            <p style="font-size:18px; color:#555;">Hello <b>{name}</b>,</p>
            <p>We are excited to announce that <b>Batch-7</b> of our <b>DevSecOps & Cloud DevOps Bootcamp</b> is starting on <b>2nd November 2024</b>. Secure your spot now!</p>
            <ul>
                <li>CI/CD Tools</li>
                <li>Infrastructure as Code</li>
                <li>Security Tools</li>
                <li>Cloud Platforms</li>
                <li>Hands-on Projects</li>
            </ul>
            <p style="text-align:center;"><a href="https://devopsshack.com" style="background-color:#4CAF50; color:white; padding:15px; border-radius:5px;">Enroll Now</a></p>
            <p>Best Regards,<br>DevOps Shack Team</p>
        </div>
    </body>
    </html>"""

    for recipient in recipients:
        name = recipient.split('@')[0].capitalize()  # Personalize the message
        html_body = html_body_template.format(name=name)

        # Create the email message
        msg = MIMEMultipart()
        msg['Subject'] = subject
        msg['From'] = sender_email
        msg['To'] = recipient
        msg.attach(MIMEText(html_body, 'html'))

        # Attach the PDF if provided
        if pdf_path and os.path.exists(pdf_path):
            with open(pdf_path, 'rb') as pdf_file:
                pdf_part = MIMEBase('application', 'octet-stream')
                pdf_part.set_payload(pdf_file.read())
                encoders.encode_base64(pdf_part)
                pdf_part.add_header('Content-Disposition', f'attachment;
filename={os.path.basename(pdf_path)}')
                msg.attach(pdf_part)

        # Send the email
        with smtplib.SMTP_SSL('smtp.gmail.com', 465) as server:
            server.login(sender_email, sender_password)
            server.sendmail(sender_email, recipient, msg.as_string())

# Example usage
recipients_list = ['recipient1@example.com', 'recipient2@example.com']
send_individual_emails('Enroll Now: Batch-7 Starting on 2nd November', 'Batch-7 is starting soon!',
recipients_list, 'Batch-7-Syllabus.pdf')
```

Script 2 | Web Scraping

```
pip install requests beautifulsoup4
```

```
import requests
from bs4 import BeautifulSoup

def scrape_headlines_demo(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    headlines = soup.find_all('h2')
    for idx, headline in enumerate(headlines, 1):
        print(f'{idx}: {headline.text.strip()}')

# Example usage
scrape_headlines_demo('https://www.indiatoday.in/india')
```

Script 3 | Automating File Handling

```
import os
import shutil

def move_files_by_type(source_dir, dest_dir, file_extension):
    for filename in os.listdir(source_dir):
        if filename.endswith(file_extension):
            shutil.move(os.path.join(source_dir, filename), os.path.join(dest_dir, filename))

# Example usage
move_files_by_type('/path/to/source', '/path/to/destination', '.txt')
```

Script 4 | Scaling AKS Resources Based on Weather

```
import requests
import subprocess

def get_weather_data(city):
    api_key = 'your_weatherapi_key'
    base_url = f'http://api.weatherapi.com/v1/current.json?key={api_key}&q={city}'
    response = requests.get(base_url)
    data = response.json()

    if 'error' not in data:
        weather = data['current']['condition']['text']
        temp_c = data['current']['temp_c']
        print(f'Weather in {city}: {weather}, Temperature: {temp_c}°C')

        if "Heavy rain" in weather:
            print("Scaling up AKS pods due to heavy rain.")
            scale_aks_pods('default', 'my-app', 3)

def scale_aks_pods(namespace, deployment_name, replicas):
    subprocess.run(['kubectl', 'scale', f'deployment/{deployment_name}', f'--replicas={replicas}', '-n',
namespace], check=True)
    print(f"Scaled {deployment_name} to {replicas} pods.")

# Example usage
get_weather_data('London')
```

Script 5 | ChatBot Interactions

```
from nltk.chat.util import Chat, reflections

pairs = [
    [r"(.*)help(.*)", ["How can I assist you today?"]],
    [r"(.*)price of (.*)", ["The price of %2 is $50."]],
    [r"(.*)course(.*)", ["Check out the DevOps course here: https://devopsshack.com"]],
    [r"quit", ["Goodbye!"]]
]

def basic_chatbot():
    print("Welcome to Customer Support! (type 'quit' to exit)")
    chat = Chat(pairs, reflections)
    chat.converse()

# Start the chatbot
basic_chatbot()
```

Script 6 | Monitor Website & Get Notified on Slack

```
import requests

def monitor_server_health(server_url, slack_webhook):
    try:
        response = requests.get(server_url)
        if response.status_code == 200:
            message = f"Server {server_url} is UP!"
        else:
            message = f"Server {server_url} is DOWN! Status Code: {response.status_code}"

        slack_data = {'text': message}
        slack_response = requests.post(slack_webhook, json=slack_data)

        if slack_response.status_code == 200:
            print(f"Slack notification sent: {message}")
        else:
            print(f"Failed to send Slack notification. Status Code: {slack_response.status_code}")
    except requests.exceptions.RequestException as e:
        print(f"Error: {e}")

# Example usage
server_url = 'https://your-server-url.com'
slack_webhook = 'https://hooks.slack.com/services/XXXX/XXXX/XXXX'
monitor_server_health(server_url, slack_webhook)
```

Script 7 | Backup in ZIP

```
import os
import zipfile
from datetime import datetime

def backup_and_zip_files(source_folder, backup_folder):
    if not os.path.exists(backup_folder):
        os.makedirs(backup_folder)

    timestamp = datetime.now().strftime('%Y-%m-%d_%H-%M-%S')
    backup_zip_filename = os.path.join(backup_folder, f"backup_{timestamp}.zip")

    with zipfile.ZipFile(backup_zip_filename, 'w', zipfile.ZIP_DEFLATED) as zipf:
        for root, dirs, files in os.walk(source_folder):
            for file in files:
                file_path = os.path.join(root, file)
                zipf.write(file_path, os.path.relpath(file_path, source_folder))

    print(f"Backup completed: {backup_zip_filename}")

# Example usage
backup_and_zip_files('/path/to/source', '/path/to/backup')
```

Script 8 | Cleanup Directory on Condition

```
import os
import time

def cleanup_old_files(directory, days_old):
    current_time = time.time()
    for filename in os.listdir(directory):
        file_path = os.path.join(directory, filename)
        if os.path.isfile(file_path):
            file_age = current_time - os.path.getmtime(file_path)
            if file_age > days_old * 86400:  # Convert days to seconds
                os.remove(file_path)
                print(f"Deleted: {filename}")

# Example usage
cleanup_old_files('/path/to/directory', 30)
```

Script 9 | Downloading YouTube Videos

```
pip install yt-dlp feedparser
```

```
import yt_dlp as youtube_dl
from feedparser import parse

def download_latest_videos(subscription_feed_url, output_folder):
    feed = parse(subscription_feed_url)
    for entry in feed.entries[:5]:
        video_url = entry.link
        ydl_opts = {'outtmpl': f'{output_folder}/%(title)s.%(ext)s', 'format': 'best'}

        with youtube_dl.YoutubeDL(ydl_opts) as ydl:
            ydl.download([video_url])
            print(f"Downloaded {video_url}")

# Example usage
download_latest_videos("https://www.youtube.com/feeds/videos.xml?channel_id=YOUR_CHANNEL_ID", 'downloads')
```

Script 10 | Synchronizing Local Repo with Remote Repo

```
import os

def clone_or_pull_repo(repo_url, local_dir):
    if not os.path.exists(local_dir):
        os.system(f"git clone {repo_url} {local_dir}")
        print(f"Cloned {repo_url} into {local_dir}")
    else:
        os.chdir(local_dir)
        os.system("git pull")
        print(f"Updated repository in {local_dir}")

# Example usage
repositories = [
    {"url": "https://github.com/your_username/repo1.git", "dir": "/path/to/repo1"},
    {"url": "https://github.com/your_username/repo2.git", "dir": "/path/to/repo2"}
]

for repo in repositories:
    clone_or_pull_repo(repo["url"], repo["dir"])
```