Optimizing DevOps Pipelines: Cost-Efficient Automation with Python

Vishw Patel-21BIT231 (Group No.: 433)

PDEU Supervisor Name: Dr. Gunjan Thakur, Assistant Professor, ICT, School of Technology

Introduction and Objectives

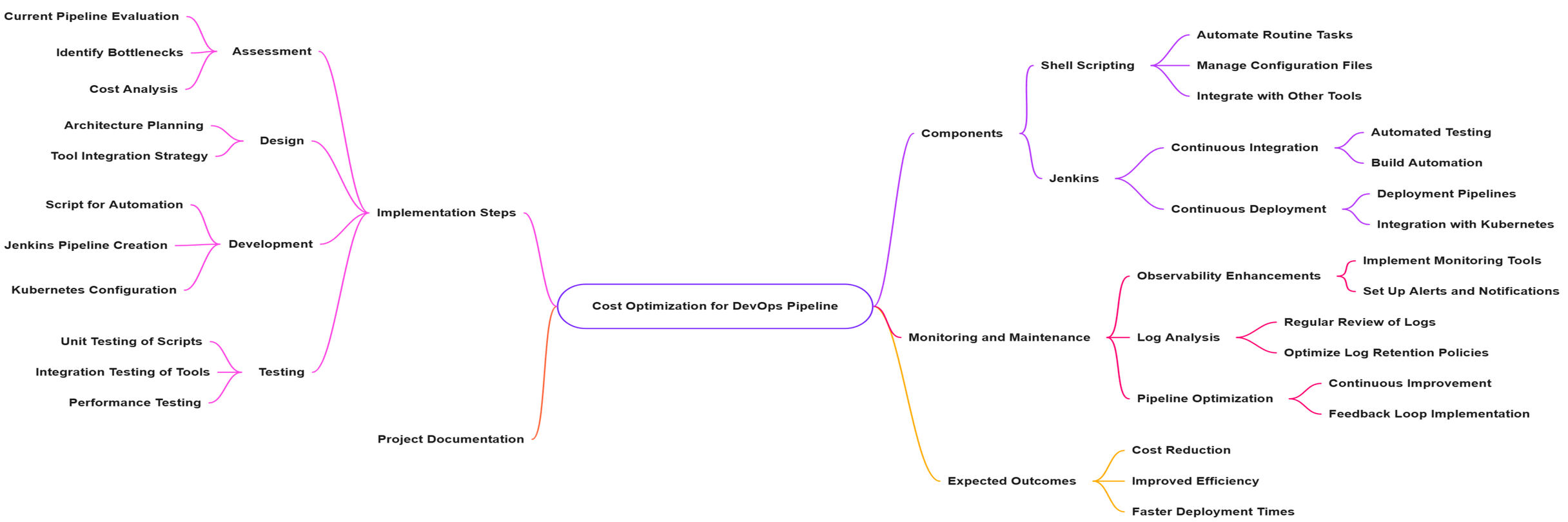
In the fast-paced world of DevOps, cost optimization plays a crucial role in ensuring efficient cloud resource utilization. Cloud resources are expensive if not managed properly.

- **To ensure a streamlined and cost-effective DevOps pipeline without compromising performance.**
- **To analyze the total cost savings and enhanced efficiency in the DevOps pipeline.**

Research Gaps

- **Model accuracy depends on the quality of training data; difficulty in real-time adaptability**
- **Trade-off between cost and performance; complexity in multi-cloud environments**
- **Limited real-world applicability due to lack of empirical validation**
- **Computational overhead in Reinforcement Learning models; scalability concerns**
- **Complexity in inter-provider coordination and SLA (Service Level Agreement) enforcement**

Methodology and Techniques



The Mind-Map outlines a structured approach to Cost Optimization for DevOps Pipelines, covering key aspects like current pipeline evaluation (bottleneck identification, cost analysis), implementation steps (automation, Jenkins, Kubernetes), monitoring and maintenance (log analysis, observability), and expected outcomes (cost reduction, efficiency, faster deployment).

Work Completed and Results

```
Enter Jenkins home directory: /var/lib/jenkins
Enter S3 bucket name (e.g., s3://my-bucket): s3://my-log-archive-bucket
2025-03-05 11:47:04 - INFO - Load resources name in resource_names.json
Resource Names Loaded:
{
    "sg-d48c29a1": "MyWebSecGroup",
    "i-09b72a51c3e8fd2": "MyWebAppInstance",
    "i-810e36b3qu8xl06":  "MyCheckInstance",
    "nat-0c5a9q6d1120": "MyNATGateway",
    "webapp-75c8k976-4q6f2": "MyWebAppPod"
}
2025-03-05 11:47:12 - INFO - Running AWS optimizations...
2025-03-05 11:47:29 - WARNING - Security Group MyWebSecGroup has an open rule! Consider restricting it. Potential Savings: $6.00
2025-03-05 11:48:05 - INFO - EC2 Instance MyWebAppInstance (t3.micro) is running. Consider rightsizing, stopping, or using reserved instances. Potential Savings: $9.43
2025-03-05 11:48:38 - INFO - EC2 Instance MyCheckInstance (t2.small) is running. Consider rightsizing, stopping, or using reserved instances. Potential Savings: $5.77
2025-03-05 11:49:22 - INFO - RDS Instance my-db-instance (mysql) is running. By using Aurora Serverless (if applicable). Potential Savings: $7.38
2025-03-05 11:49:59 - INFO - NAT Gateway MyNATGateway might be costing you money! Use Transit Gateway. Potential Savings: $8.10
2025-03-05 11:50:17 - INFO - Uploaded job1/100 to s3://my-log-archive-bucket/job1-100.log.gz. Savings: $0.05
2025-03-05 11:50:21 - INFO - Uploaded job2/200 to s3://my-log-archive-bucket/job2-200.log.gz. Savings: $0.07
2025-03-05 11:50:23 - INFO - Uploaded job3/300 to s3://my-log-archive-bucket/job3-300.log.gz. Savings: $0.03
2025-03-05 11:50:29 - INFO - Uploaded job4/400 to s3://my-log-archive-bucket/job4-400.log.gz. Savings: $0.09
2025-03-05 11:51:02 - INFO - Pod MyWebAppPod in namespace default is requesting 1.00 CPU cores and 1.00 GB memory. Consider right-sizing. Potential Savings: $7.26
2025-03-05 11:51:17 - INFO - Pod my-pod-2 in namespace default is requesting 0.50 CPU cores and 0.50 GB memory. Consider right-sizing. Potential Savings: $3.61
2025-03-05 11:52:01 - INFO -
Total Initial Estimated Cost: $78.26
Total Estimated Cost Savings: $30.47
Optimized cost: $47.79
Overall Improved Efficiency: 38.93%
2025-03-05 11:52:14 - INFO - Saved resource names to resource_names.json
```

(a) Optimized Cost for AWS

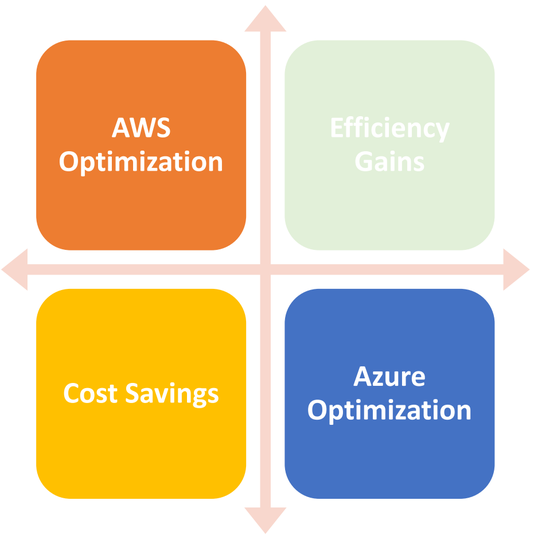
```
Enter Jenkins home directory: /var/lib/jenkins
Enter Azure Storage Account name: my_storage_account
Enter Azure Blob Container name: jenkins-logs
2025-03-17 14:32:21 - INFO - Load resource names from resource_names.json
Resource Names Loaded:
{
"myWebSecGroup": "myWebSecGroup-sg",
"myWebAppInstance": "mywebappinstance-vm",
"myCheckInstance": "mycheckinstance-vm",
"myNatGateway": "myNatGateway-nat",
"myWebAppPod": "mywebapppod-aks"
}
2025-03-17 14:33:06 - INFO - Running Azure optimizations...
2025-03-17 14:33:14 - INFO - Calling optimize_vms...
2025-03-17 14:33:19 - INFO - Virtual Machine mywebappinstance-vm (Standard_B1ms) is running. Consider rightsizing, stopping, or using reserved instances. Potential Savings: $9.34
2025-03-17 14:33:28 - INFO - Virtual Machine mycheckinstance-vm (Standard_B1s) is running. Consider rightsizing, stopping, or using reserved instances. Potential Savings: $9.33
2025-03-17 14:33:31 - INFO - optimize_vms completed.
2025-03-17 14:33:42 - INFO - Calling optimize_sql_databases...
2025-03-17 14:33:51 - INFO - Azure SQL Database my-db-instance is running. Consider using serverless compute tier. Potential Savings: $6.06
2025-03-17 14:34:01 - INFO - optimize_sql_databases completed.
2025-03-17 14:34:05 - INFO - Calling optimize_sgs...
2025-03-17 14:34:16 - WARNING - Network Security Group myWebSecGroup-sg has an open rule! Consider restricting it. Potential Savings: $5.39
2025-03-17 14:34:21 - INFO - optimize_sgs completed.
2025-03-17 14:34:24 - INFO - Calling process_jenkins_logs...
2025-03-17 14:34:30 - INFO - Uploaded job1/100 to blob job1-100.log.gz. Savings: $0.03
2025-03-17 14:34:32 - INFO - Uploaded job2/200 to blob job2-200.log.gz. Savings: $0.09
2025-03-17 14:34:36 - INFO - Uploaded job3/300 to blob job3-300.log.gz. Savings: $0.07
2025-03-17 14:34:42 - INFO - process_jenkins_logs completed.
2025-03-17 14:34:56 - INFO - Calling optimize_aks_resources...
2025-03-17 14:35:16 - INFO - Kubernetes Pod mywebapppod-aks in namespace default is requesting 1.00 CPU cores and 1.00 GB memory. Consider right-sizing. Potential Savings: $6.85
2025-03-17 14:35:24 - INFO - Kubernetes Pod my-pod-2 in namespace default is requesting 0.50 CPU cores and 0.50 GB memory. Consider right-sizing. Potential Savings: $5.44
2025-03-17 14:35:33 - INFO - optimize_aks_resources completed.

2025-03-17 14:35:46 - INFO -
Total Initial Estimated Cost: $72.03
Total Estimated Cost Savings: $29.43
Optimized cost: $42.60
Overall Improved Efficiency: 40.86%
2025-03-17 14:35:55 - INFO - Azure optimizations completed.
```

(b) Optimized Cost for Azure

- **The AWS optimization reduced costs from $78.26 to $47.79, achieving 38.93% improved efficiency, while the Azure optimization lowered costs from $72.03 to $42.60, improving efficiency by 40.86%.**

Conclusion



The cost optimization analysis for AWS and Azure revealed significant opportunities for reducing expenses and improving efficiency. AWS achieved 38.93% efficiency improvement, while Azure slightly outperformed with 40.86% efficiency improvement. This structured approach ensures long-term financial and operational benefits in cloud-based DevOps pipelines.

Bibliography/ References

https://thesai.org/Publications/ViewPaper?Code=IJACSA&Issue=4&SerialNo=20&Volume=6
https://dl.acm.org/doi/10.1145/3582883
https://www.sciencedirect.com/science/article/pii/S0167739X15002800
https://www.computer.org/csdl/journal/cc/2022/03/09165211/1mcQUWRgS6A
https://www.computer.org/csdl/journal/sc/2012/02/tsc2012020164/13rRUxBa53q
http://aws.amazon.com/
https://azure.microsoft.com/