ADDITIONAL TOPICS

**Building Serverless Solutions
with Azure and .NET**

Scaling and Performance Optimization

Maximizing Efficiency in Serverless Solutions

Scaling in Serverless

Concurrency Management

Provisioned Concurrency

Cold Starts

Optimizing Performance

Efficient Code Design	**Resource Allocation**	**Managing Dependencies**
Data Handling	**Monitoring and Logging**	**Cold Start Mitigation**

Efficient Code Design

Lightweight Functions

Minimize Dependencies

Reduce Execution Time

Resource Allocation

Allocate appropriate memory and CPU

Balance performance and cost

Managing Dependencies

Bundle only necessary libraries

Use Dependency Injection

Data Handling

Optimize data retrieval and storage

Implement caching strategies

Prefer asynchronous processing

Monitoring and Logging

Use built-in monitoring tools

Implement structured logging

Cold Start Mitigation

<div style="border: 2px solid; background-color: #E8883A; border-radius: 15px; text-align: center;">

Use provisioned concurrency

</div>

<div style="border: 2px solid; background-color: #E8883A; border-radius: 15px; text-align: center;">

Leverage function warming strategies

</div>

Best Practices

Use Auto-Scaling Features

Test Thoroughly

Cost Management

Conclusion

- Scaling and performance optimization are key for fast, reliable, cost-effective applications

- Implement best practices for efficient serverless applications.

Event-Driven Mastery

Designing Flexible and Scalable Systems