Latency chaos strategy **CHEAT SHEET**

Basics

This proactive chaos **strategy injects delay** to simulate slow network, long processing.

You can configure the behaviour of the strategy via the **ChaosLatencyStrategyOptions** object.

Specify delay - short form

```
new ResiliencePipelineBuilder()
  .AddChaosLatency(0.2,
    TimeSpan.FromSeconds(10))
```

Specify delay – long form

```
new ResiliencePipelineBuilder()
  .AddChaosLatency(new ChaosLatencyStrategyOptions
  {
    Latency = TimeSpan.FromSeconds(10),
    InjectionRate = 0.2
  })
```

Specify delay dynamically with switch expression

```
new ResiliencePipelineBuilder()
  .AddChaosLatency(new ChaosLatencyStrategyOptions
  {
    LatencyGenerator = static _ =>
    {
      var rnd = Random.Shared.NextDouble();
      TimeSpan ts = rnd switch
      {
        < 0.4 => TimeSpan.FromMilliseconds(750),
        >= 0.4 and < 0.8 => TimeSpan.FromSeconds(5),
        _ => TimeSpan.Zero
      };
      return new ValueTask<TimeSpan>(ts);
    }
  })
```

Specify asynchronous delegate for injection notification

```
new ResiliencePipelineBuilder()
  .AddChaosLatency(new ChaosLatencyStrategyOptions
  {
    OnLatencyInjected = static async args => await NotifyAsync(args.Latency)
  })
```