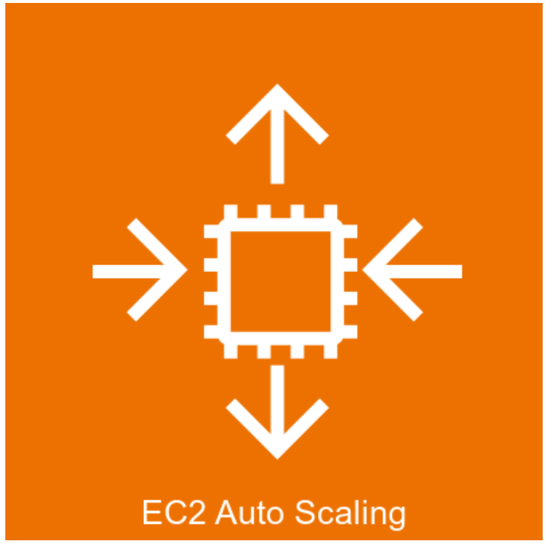
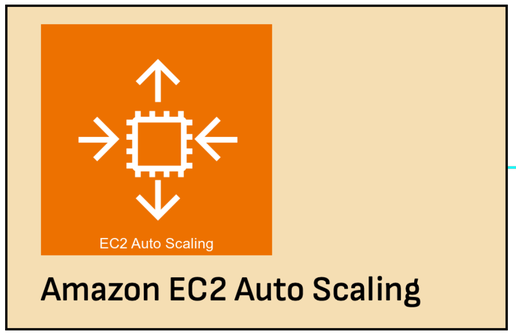


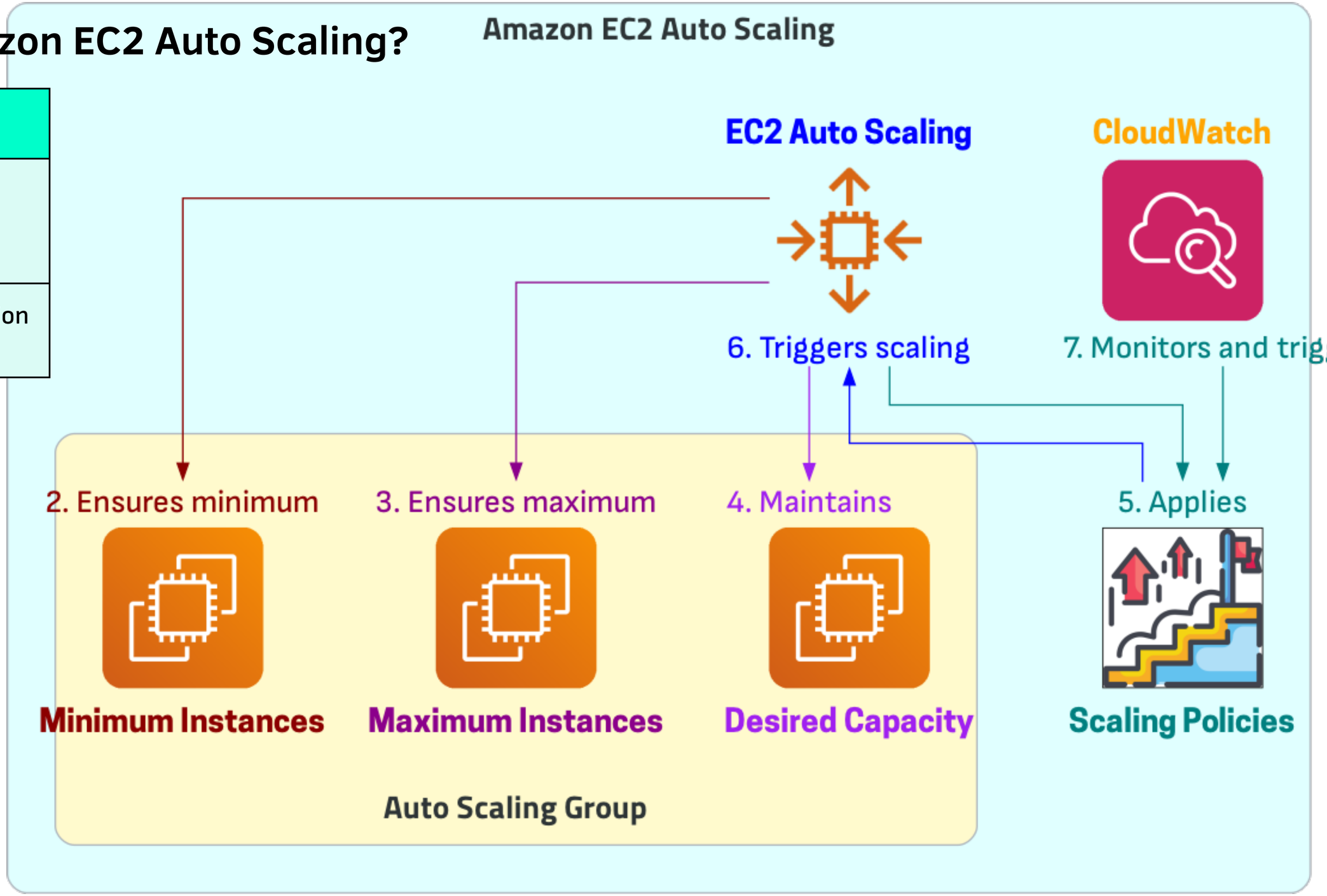
Amazon EC2 Auto Scaling

Table of Contents



What is Amazon EC2 Auto Scaling?

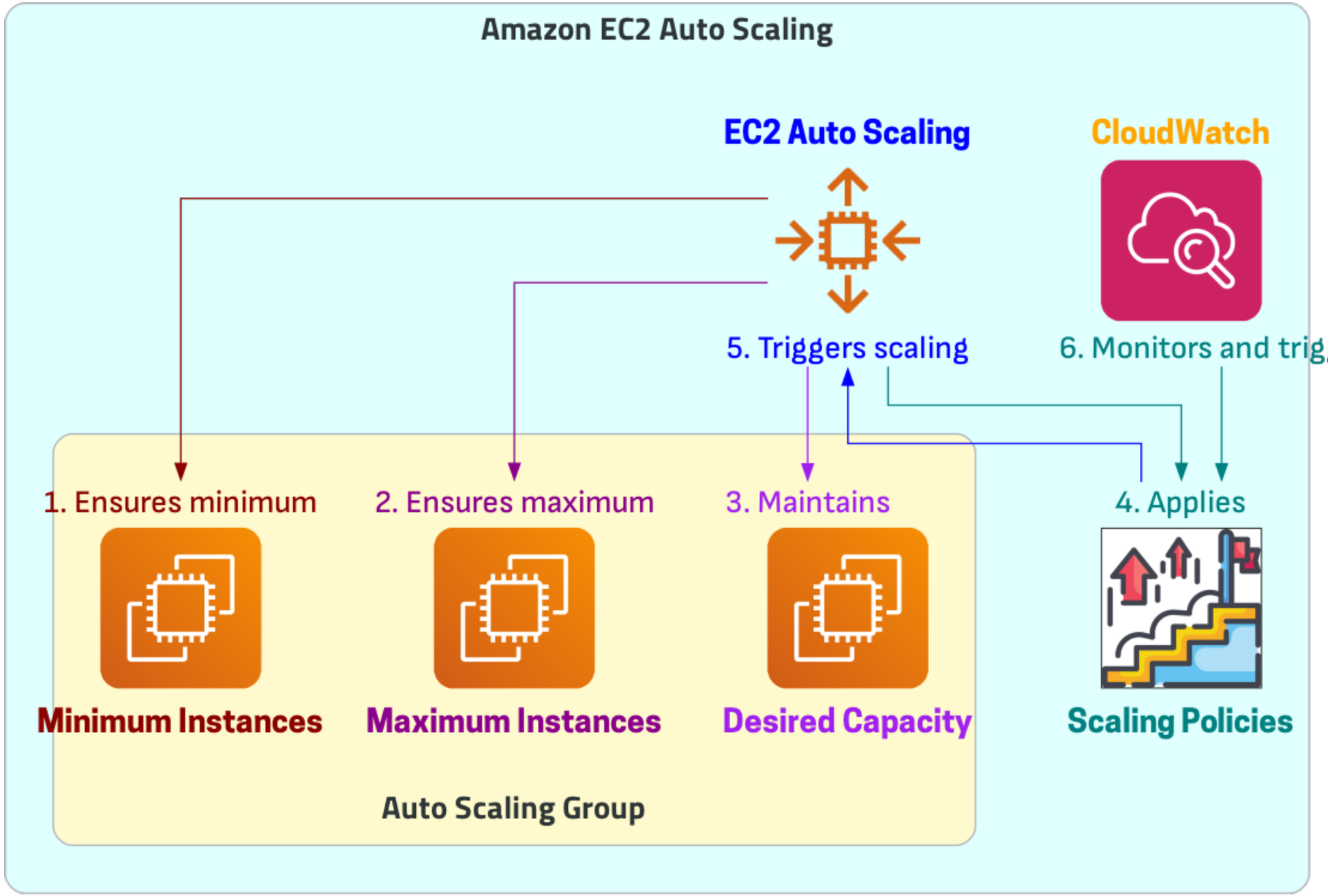
1️⃣. 🎯 Purpose
✅ Ensures correct number of EC2 instances
📊 Handles application load



What is Amazon EC2 Auto Scaling?

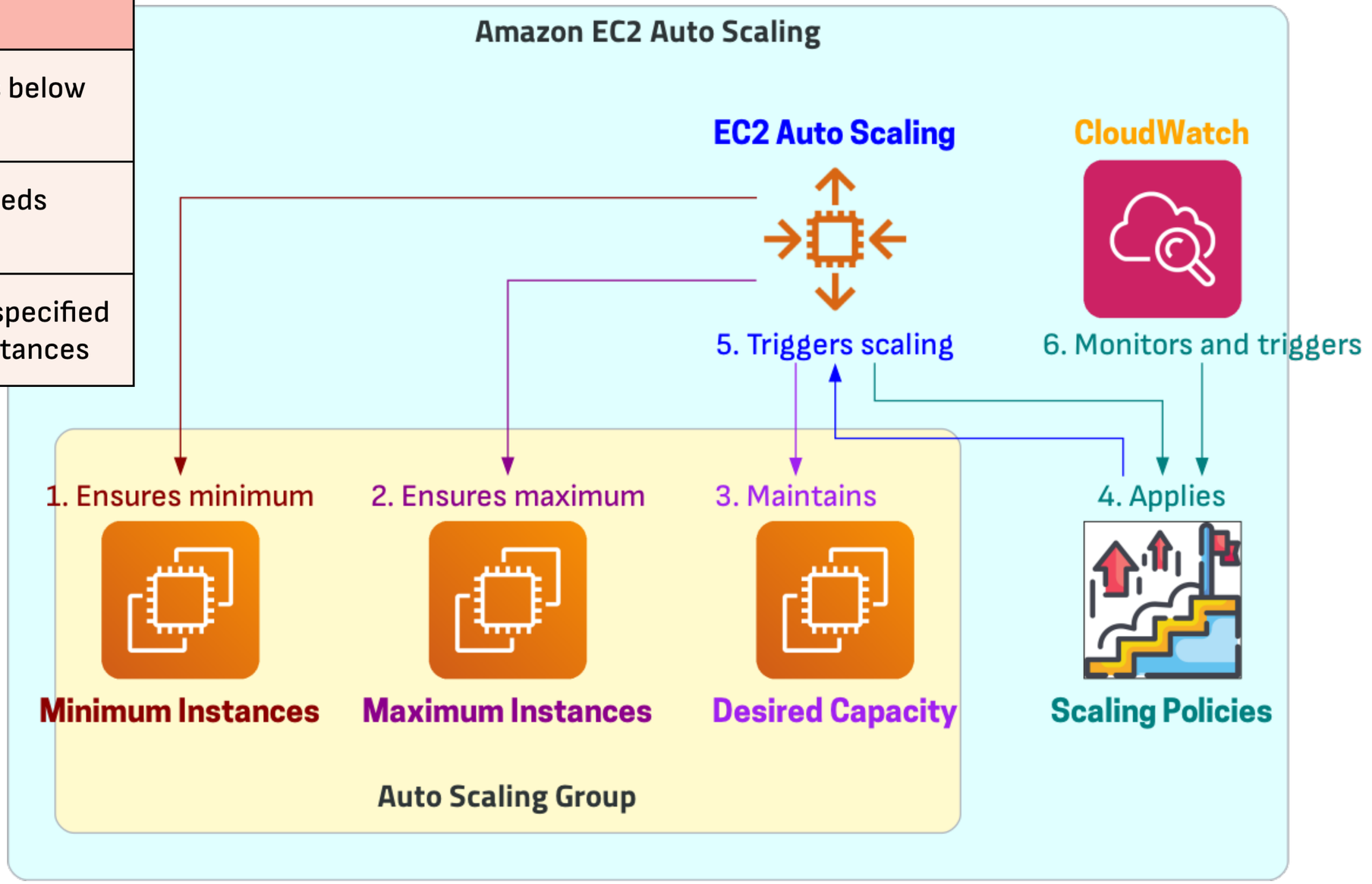
2️⃣. 🗃️ Auto Scaling groups

📦 Collections of EC2 instances



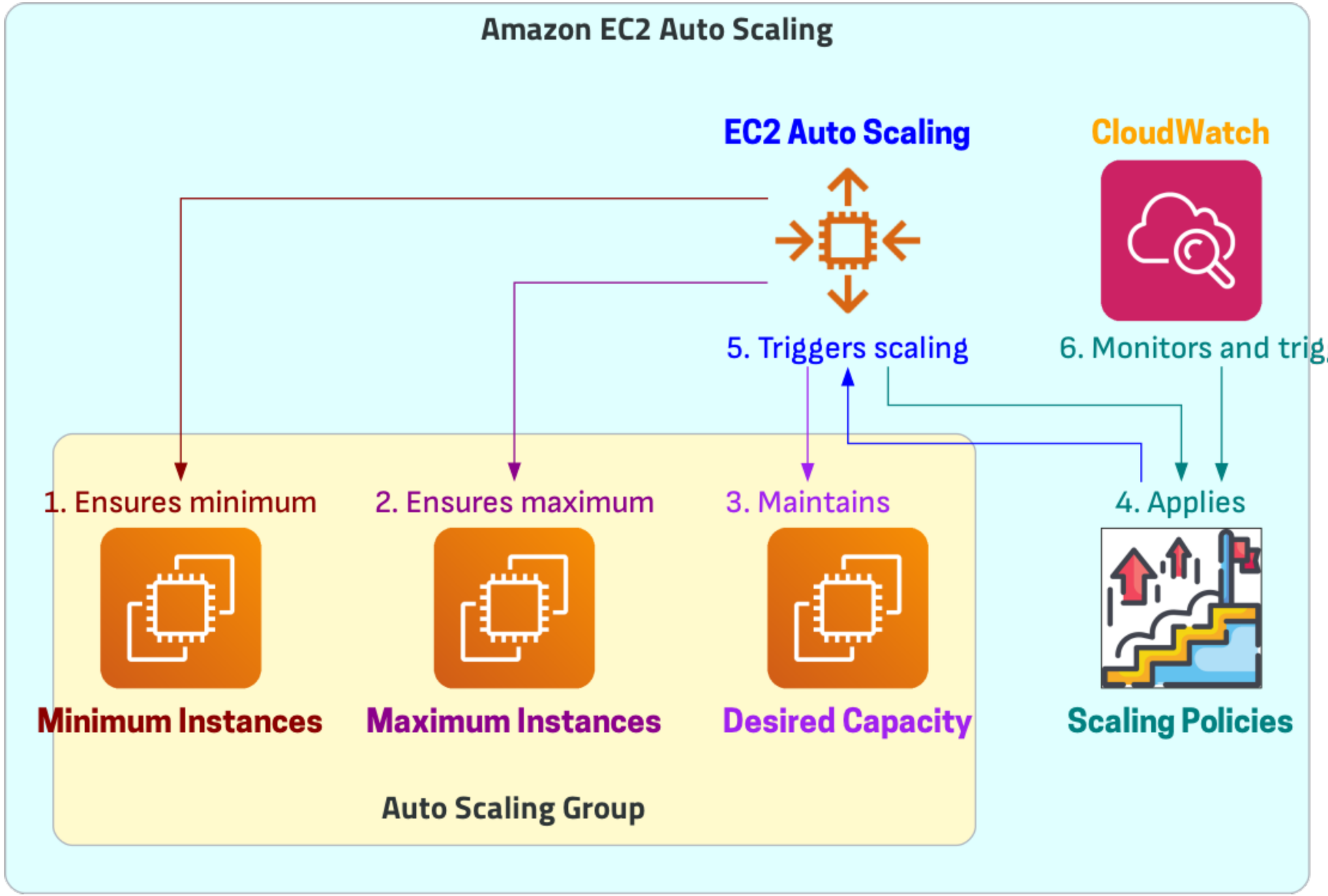
What is Amazon EC2 Auto Scaling?

3️⃣. 📊 Group size management	
🔽 Minimum instances	🛡️ Never goes below specified size
🔼 Maximum instances	🚫 Never exceeds specified size
⚖️ Desired capacity	🎛️ Maintains specified number of instances

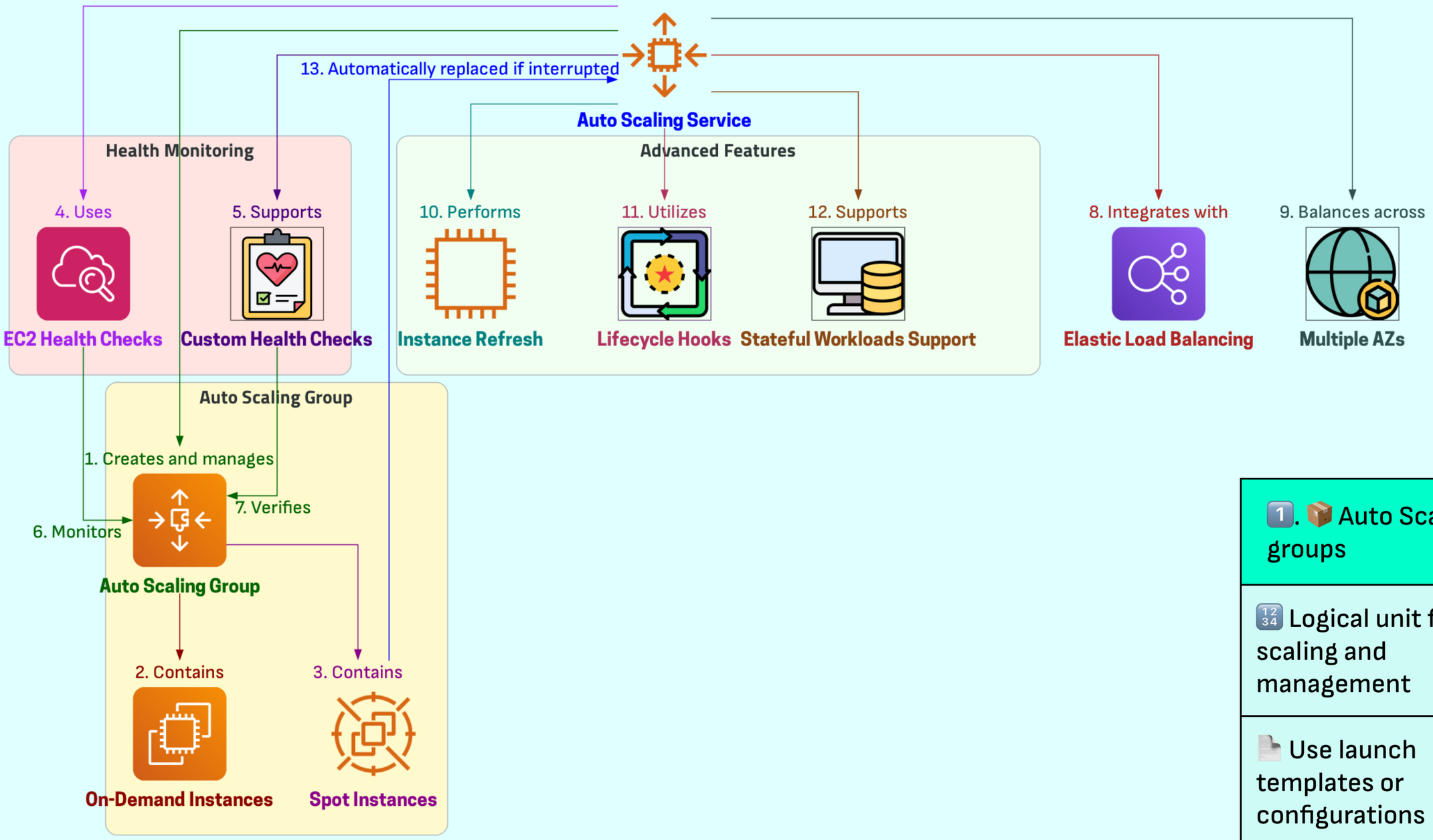


What is Amazon EC2 Auto Scaling?

4️⃣. 📈 Scaling policies
🚀 Launches instances when demand increases
🛑 Terminates instances when demand decreases



Features

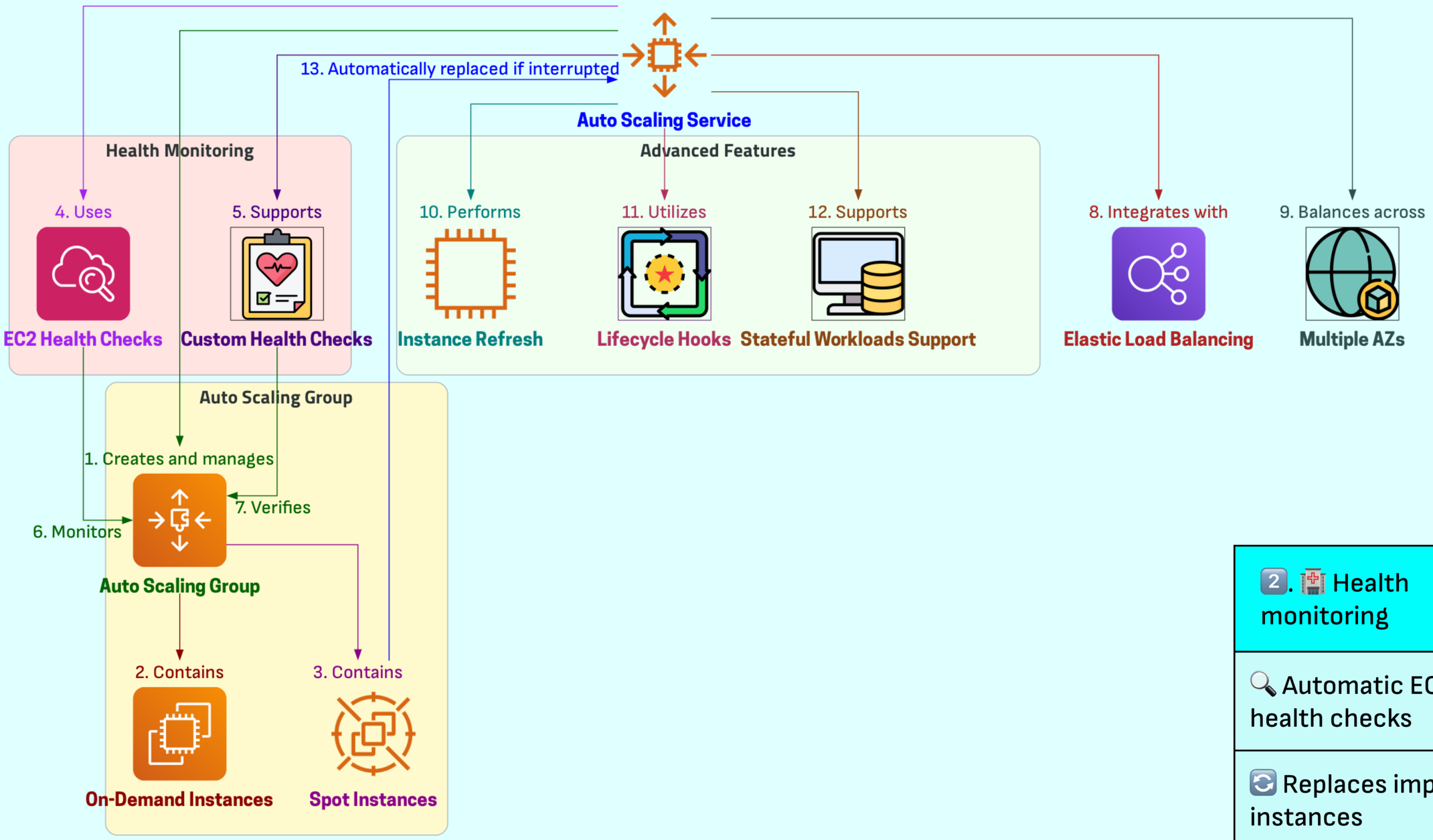


1️⃣. 📦 Auto Scaling groups

🔢 Logical unit for scaling and management

📄 Use launch templates or configurations

Features

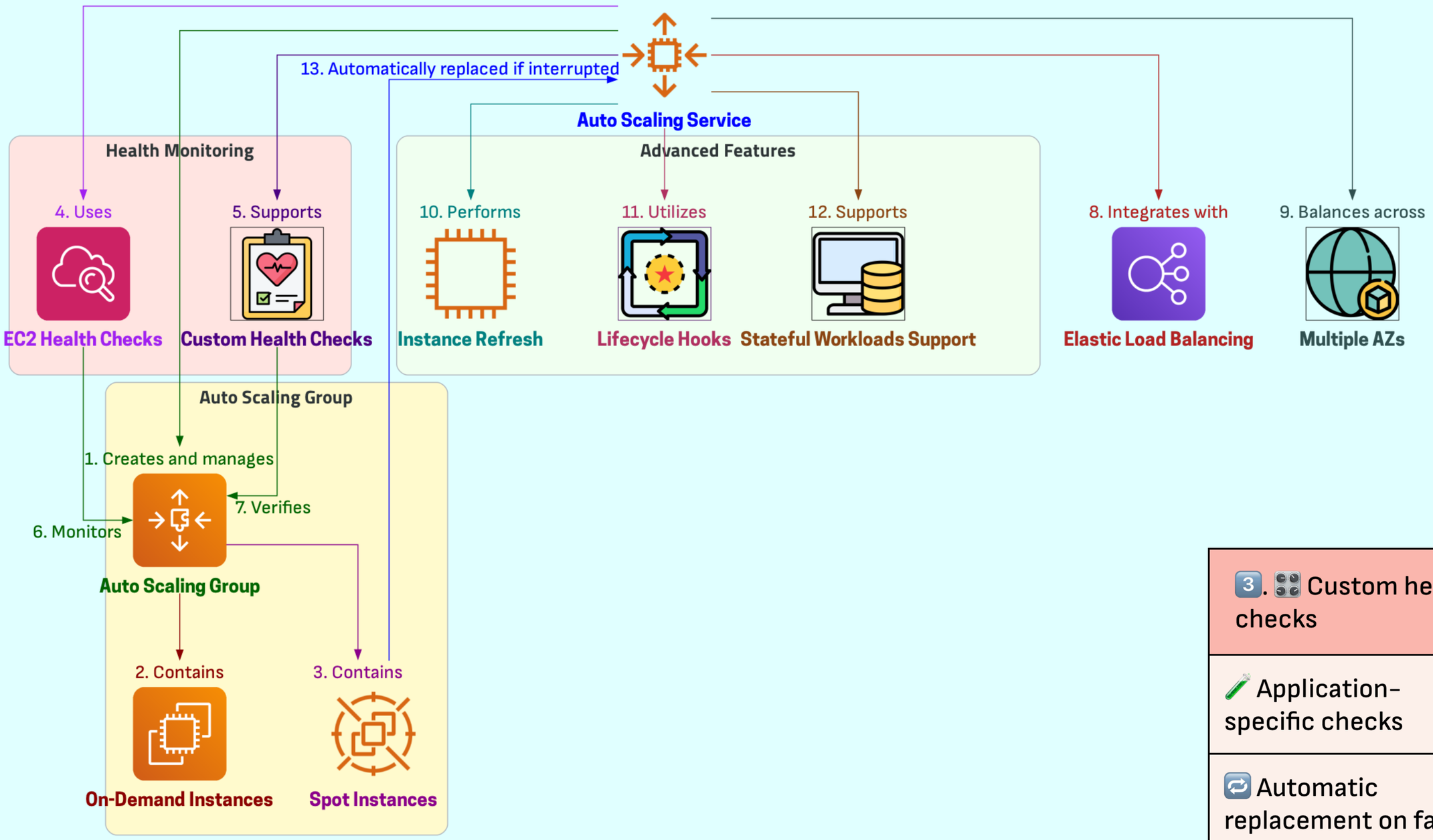


2️⃣. 🏥 Health monitoring

🔍 Automatic EC2 health checks

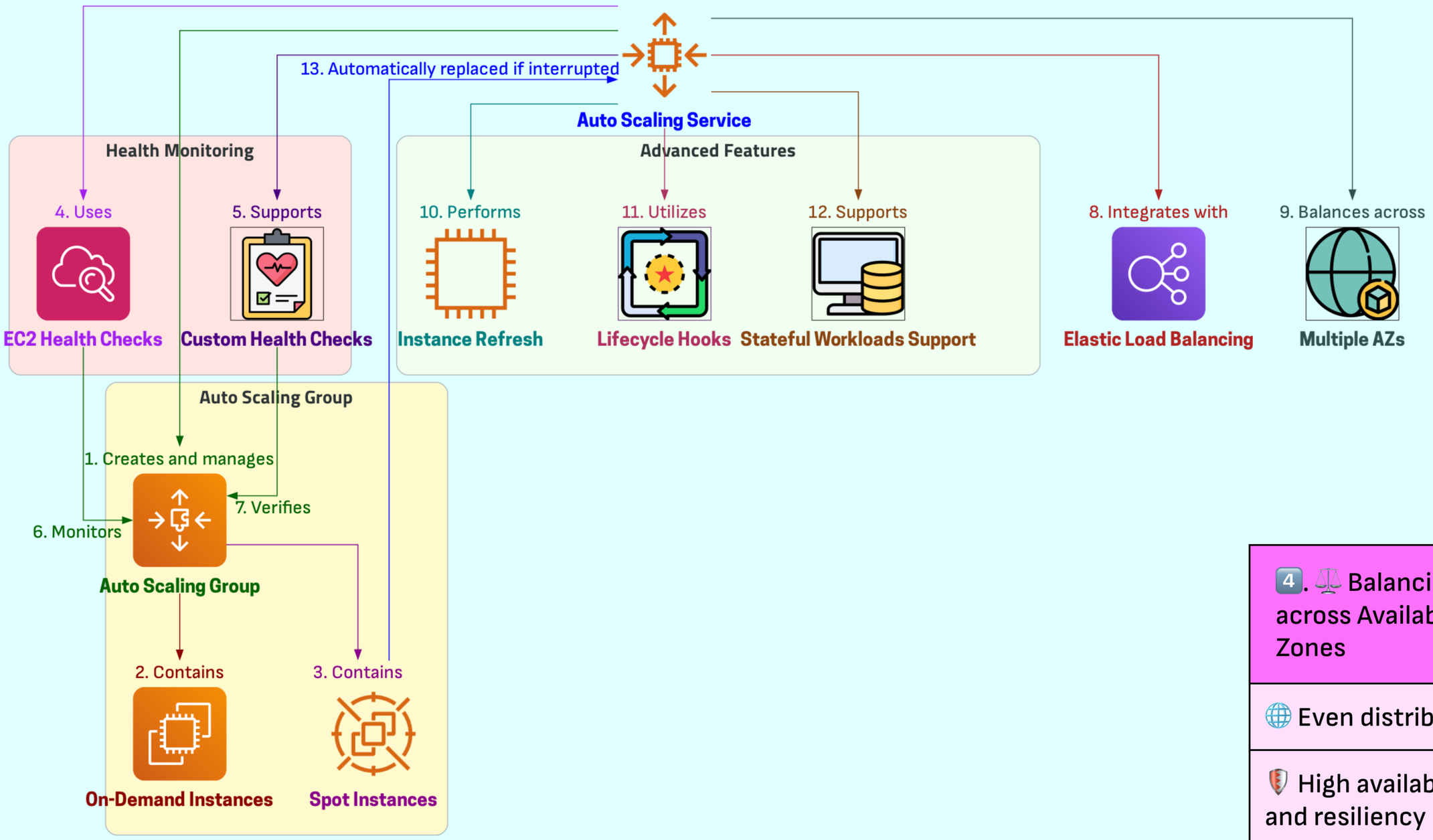
🔄 Replaces impaired instances

Features



3️⃣. 🎛️ Custom health checks
🧪 Application-specific checks
🔁 Automatic replacement on failure

Features

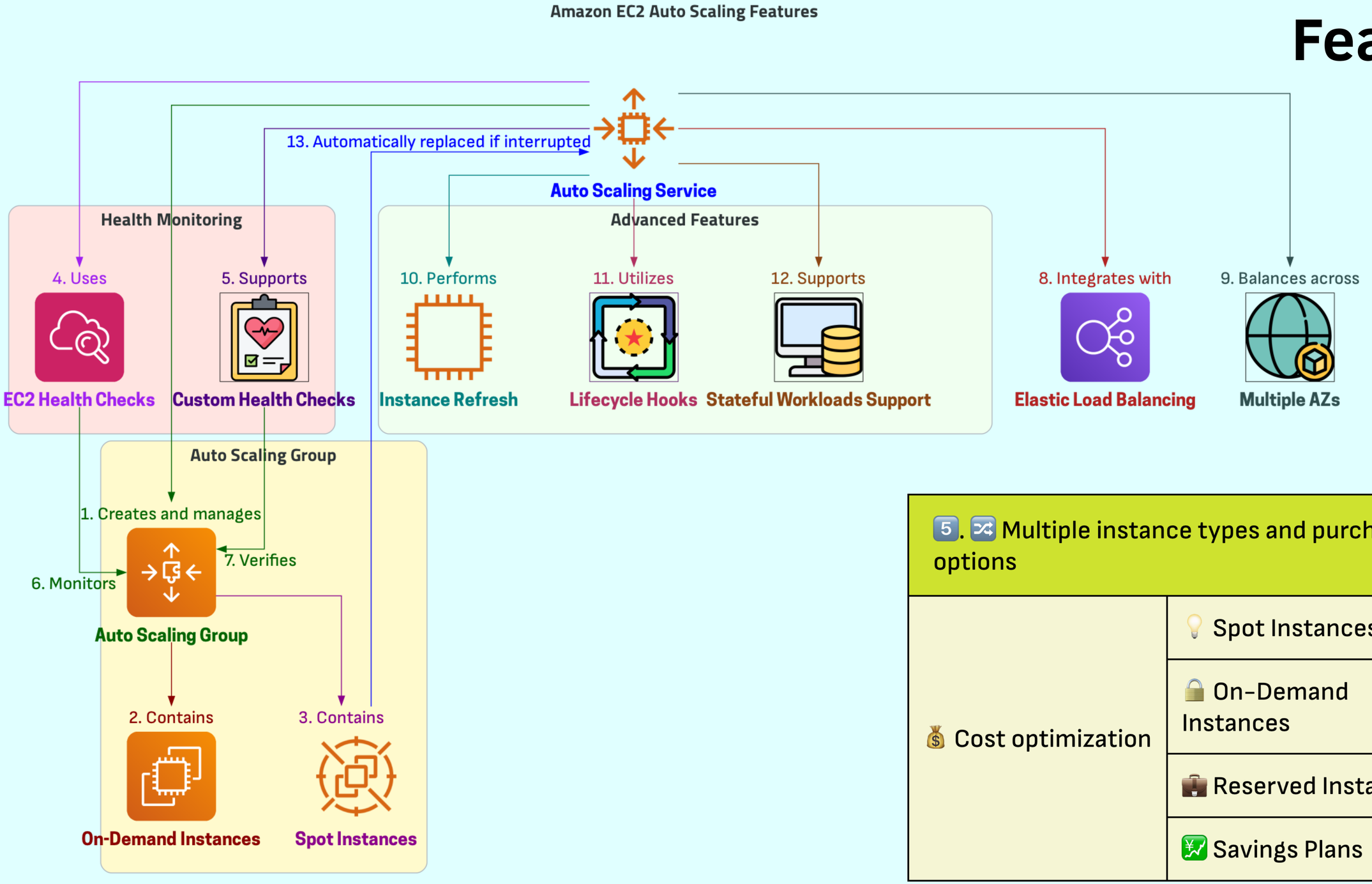


4️⃣. ⚖️ Balancing across Availability Zones

🌐 Even distribution

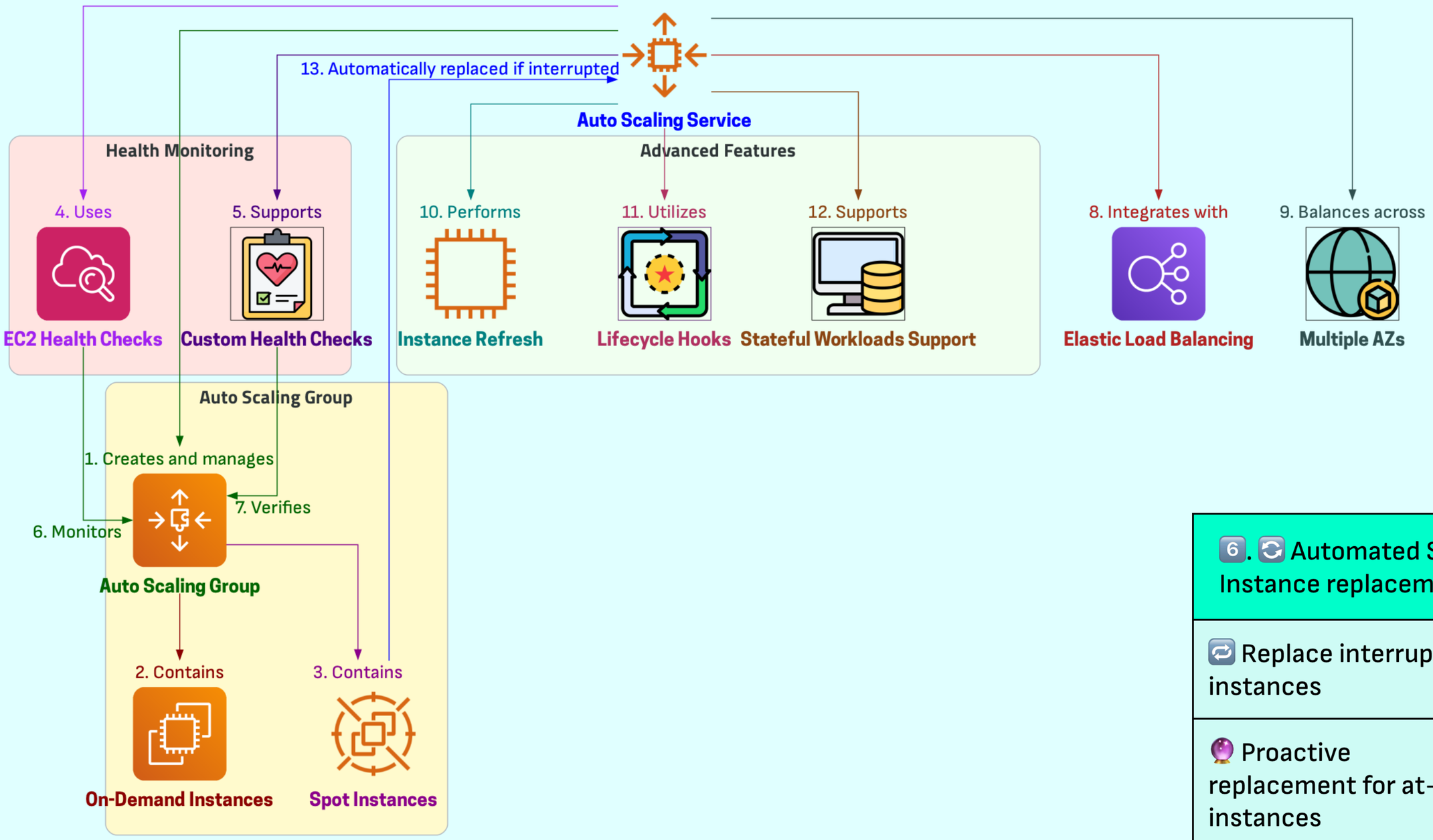
🛡️ High availability and resiliency

Features



5️⃣. 🔀 Multiple instance types and purchase options	
💰 Cost optimization	💡 Spot Instances
	🔒 On-Demand Instances
	💼 Reserved Instances
	💹 Savings Plans

Features

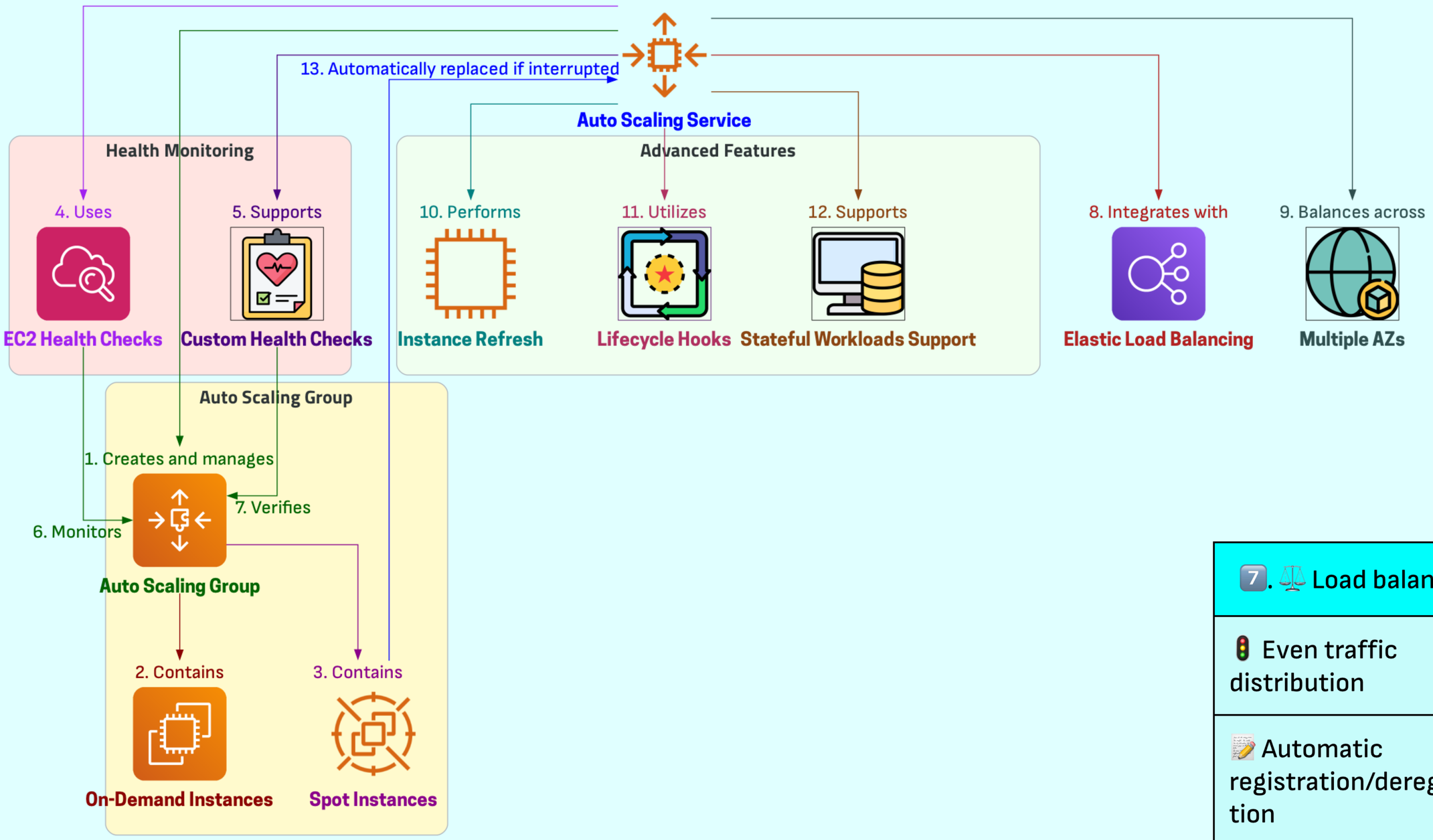


6️⃣. 🔄 Automated Spot Instance replacement

🔁 Replace interrupted instances

🔮 Proactive replacement for at-risk instances

Features

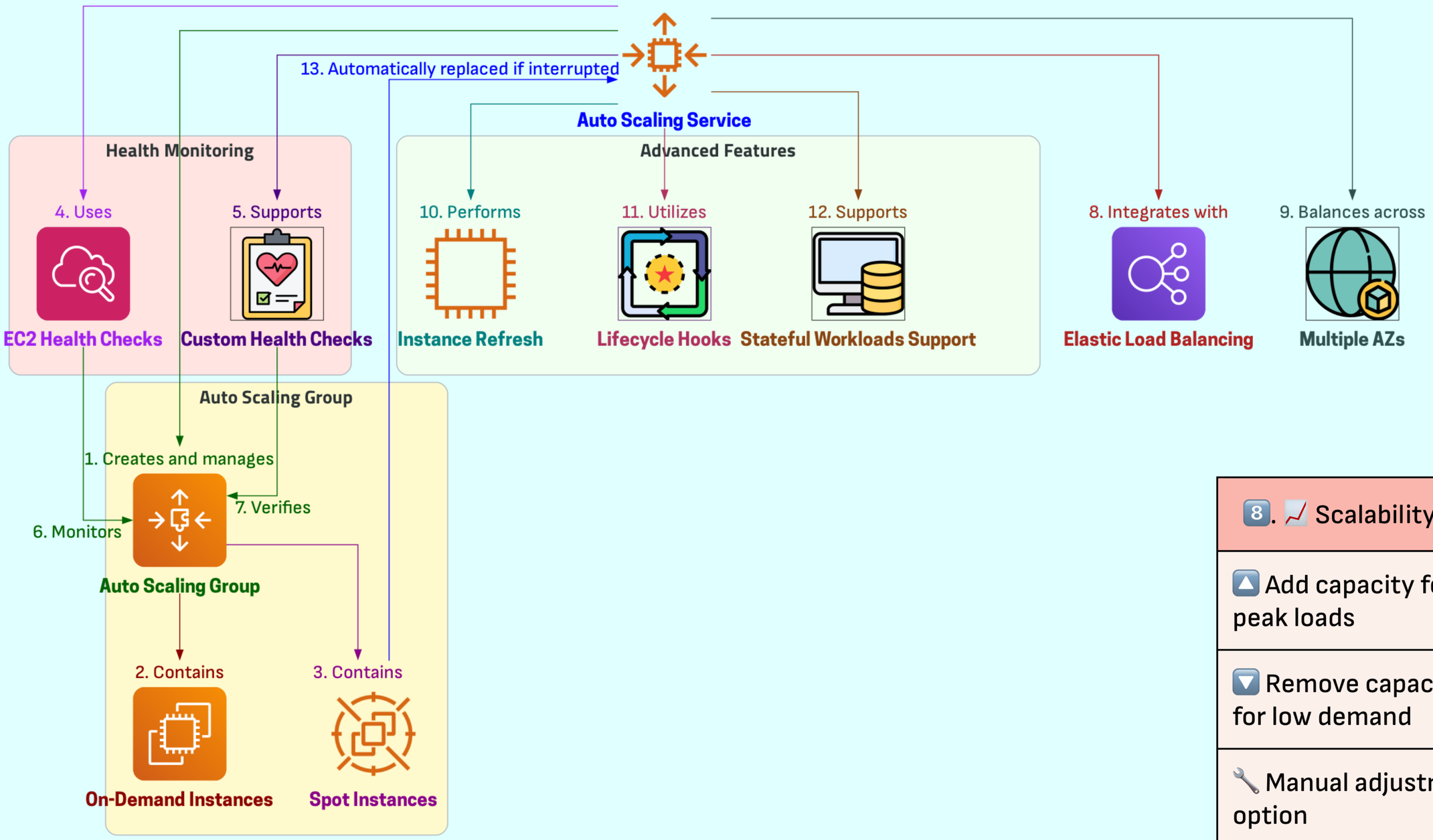


7️⃣. ⚖️ Load balancing

🚦 Even traffic distribution

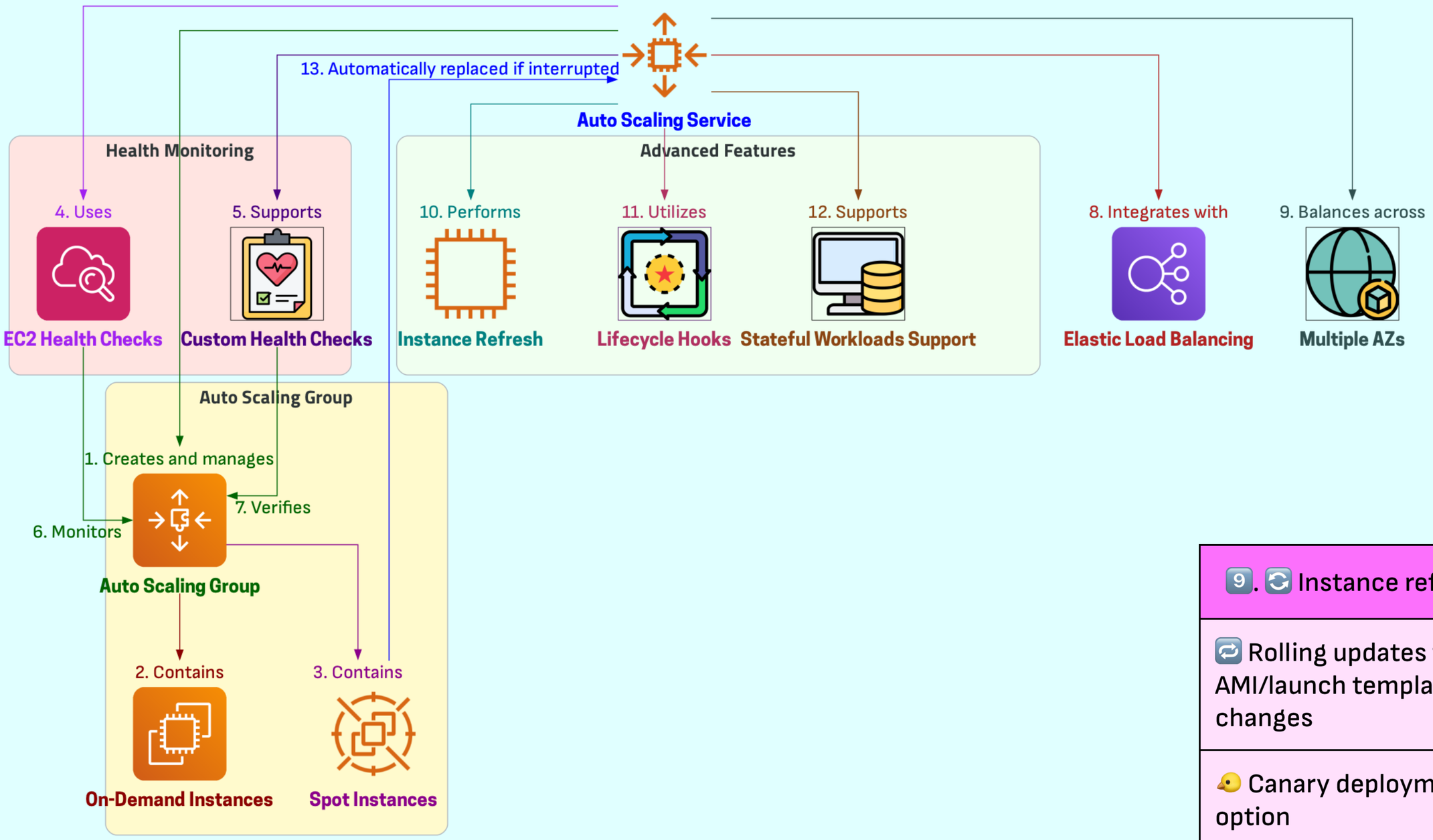
📝 Automatic registration/deregistration

Features



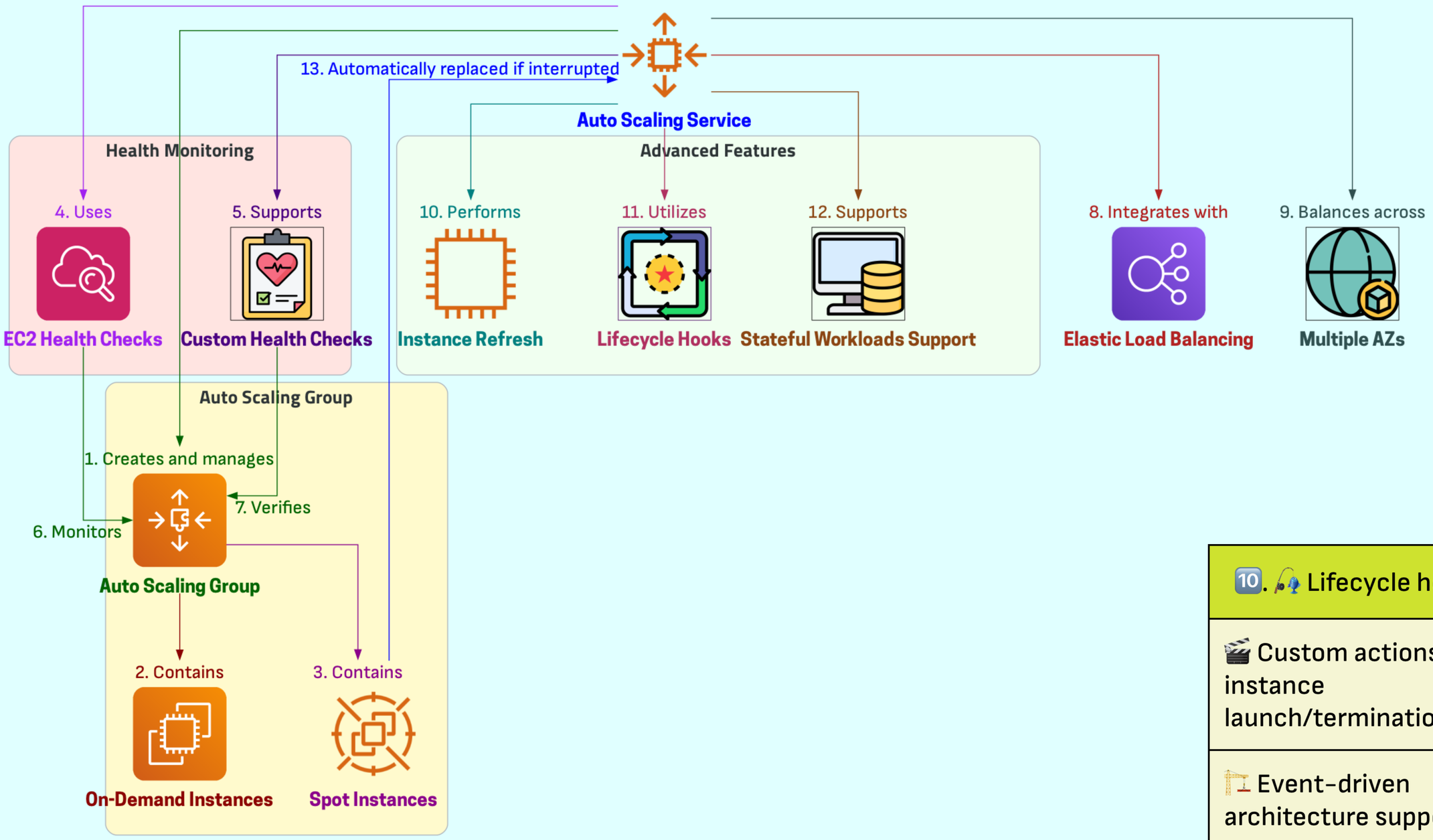
8. 📈 Scalability
🔼 Add capacity for peak loads
🔽 Remove capacity for low demand
🔧 Manual adjustment option

Features



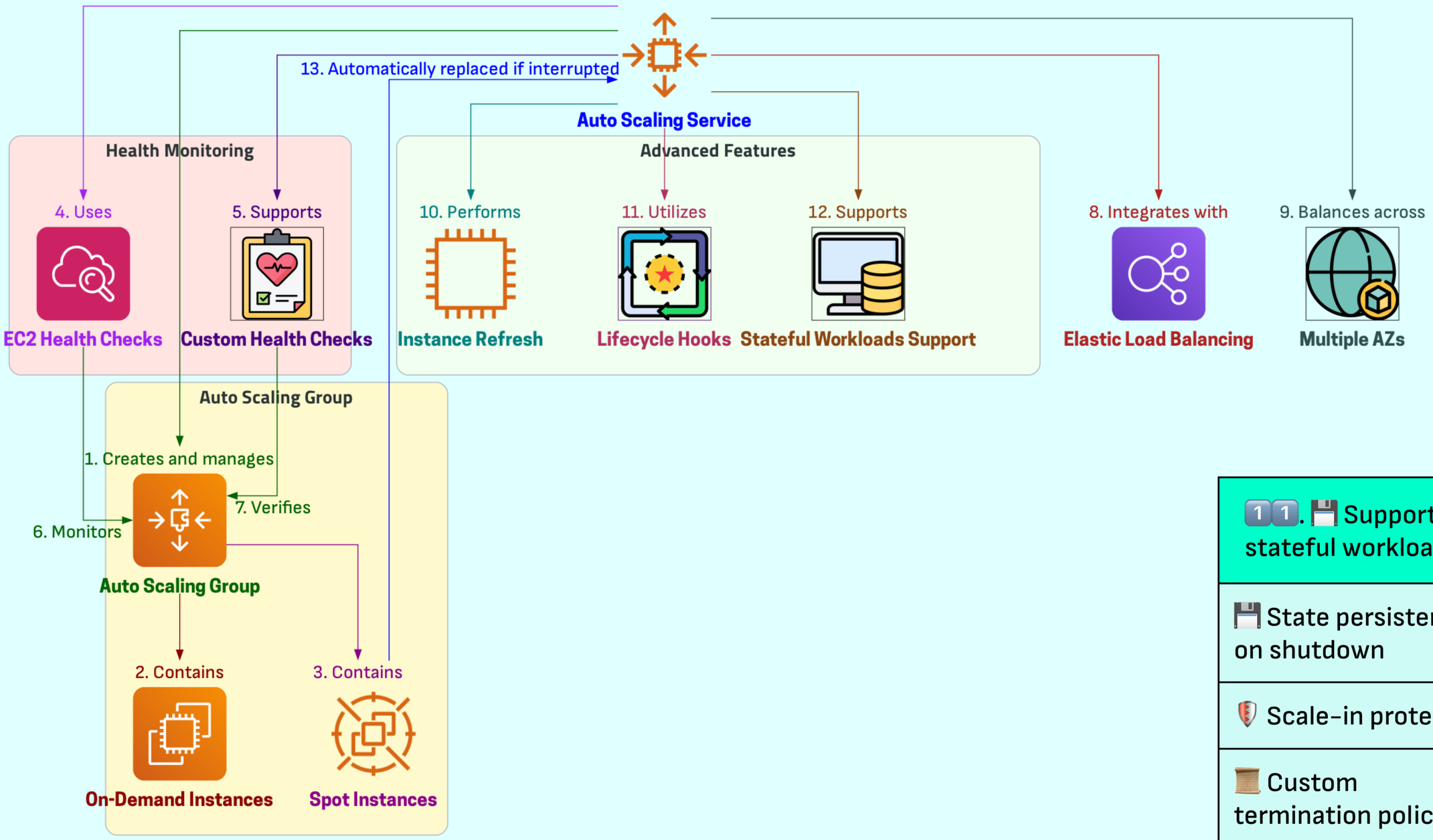
9️⃣. 🔄 Instance refresh
🔁 Rolling updates for AMI/launch template changes
🐤 Canary deployment option

Features



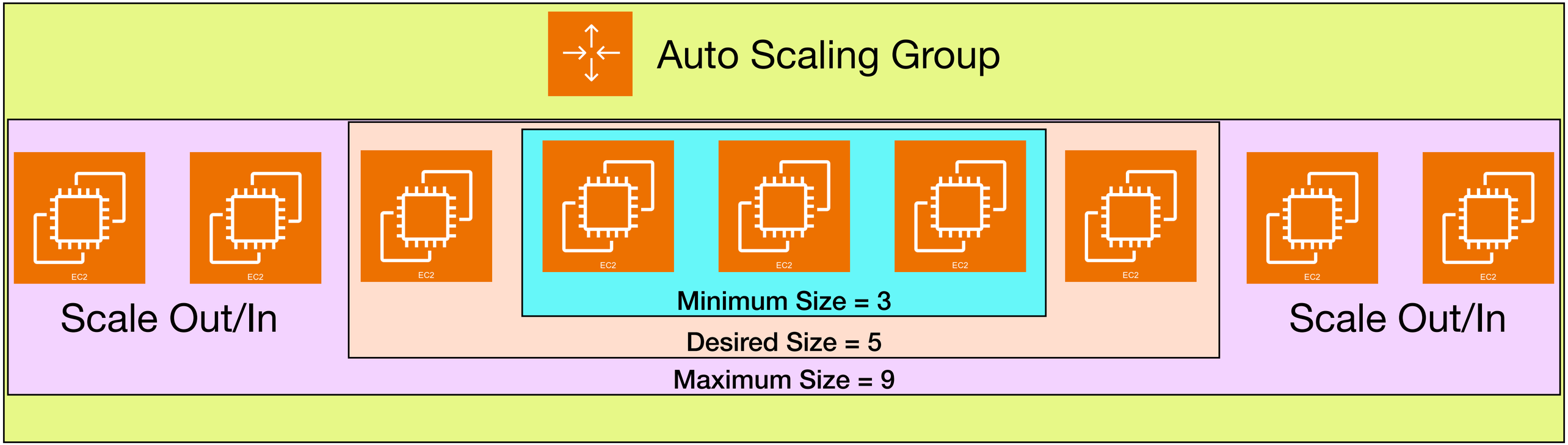
🔟. 🎣 Lifecycle hooks
🎬 Custom actions on instance launch/termination
🏗️ Event-driven architecture support

Features

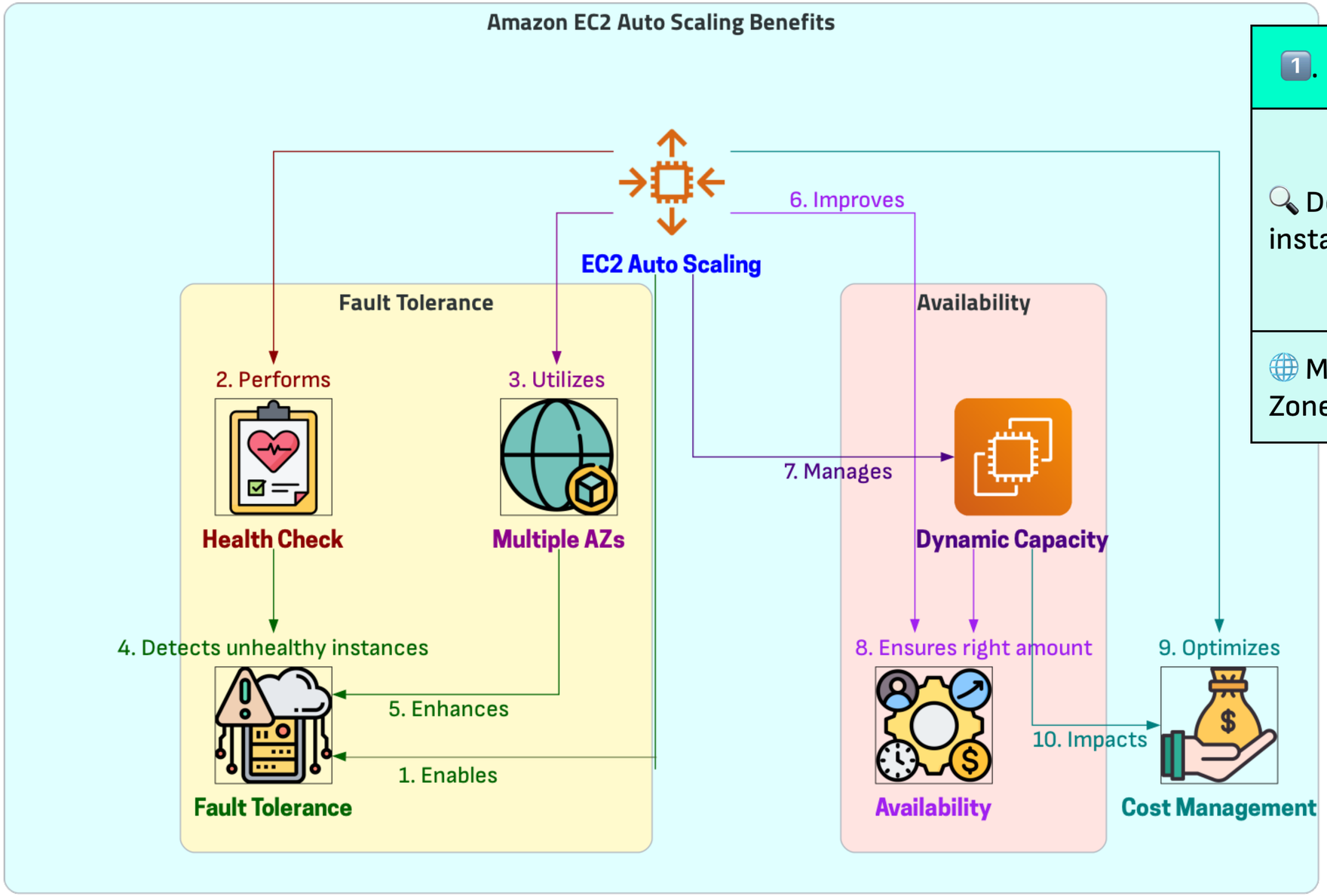


1️⃣1️⃣. 💾 Support for stateful workloads
💾 State persistence on shutdown
🛡️ Scale-in protection
📜 Custom termination policies

Auto Scaling Group Example

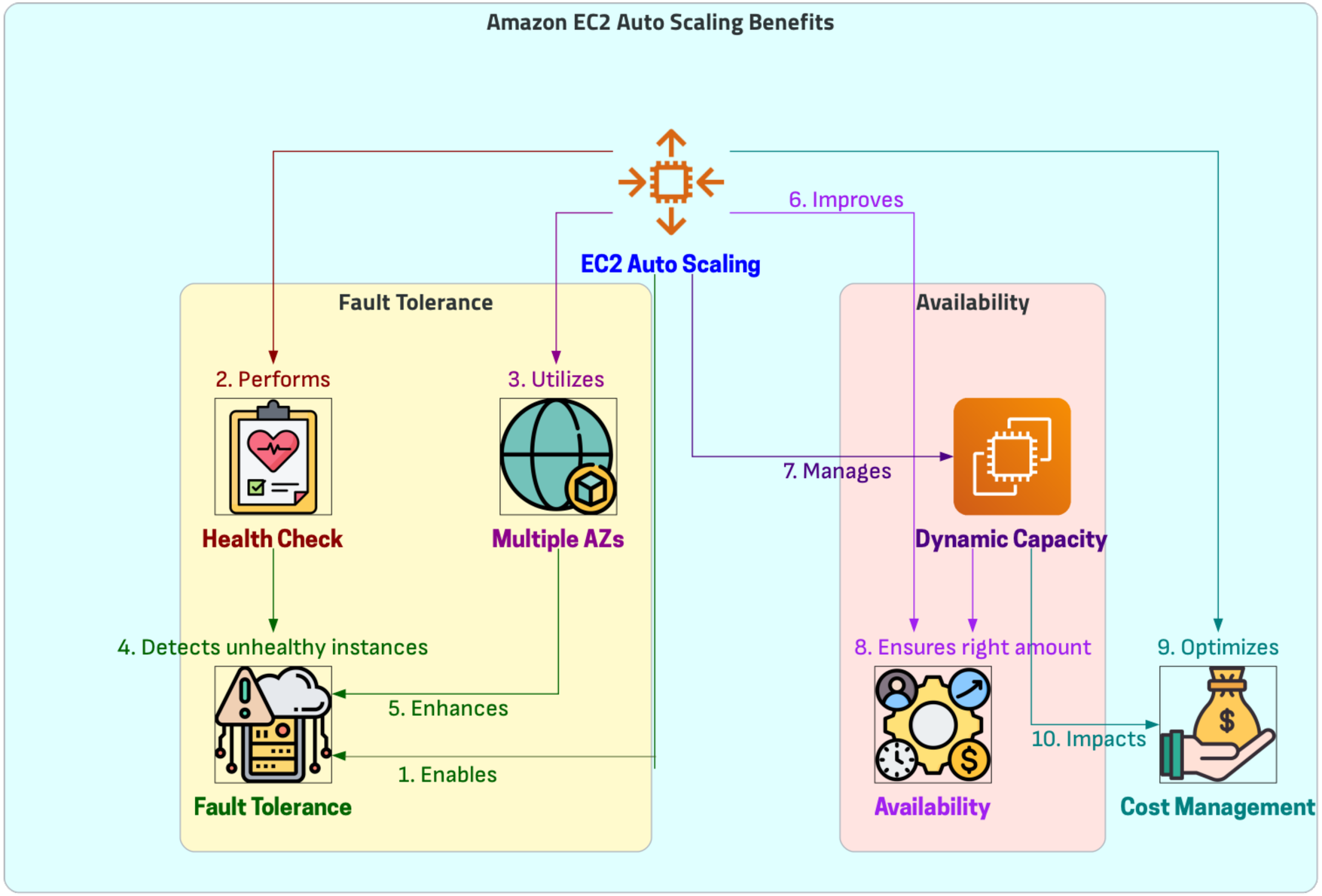


Amazon EC2 Auto Scaling Benefits



1️⃣. 🛡️ Better fault tolerance	
🔍 Detects unhealthy instances	❌ Terminates unhealthy instances
	✅ Launches replacement instances
🌐 Multiple Availability Zones	🔄 Compensates for unavailable zones

Amazon EC2 Auto Scaling Benefits

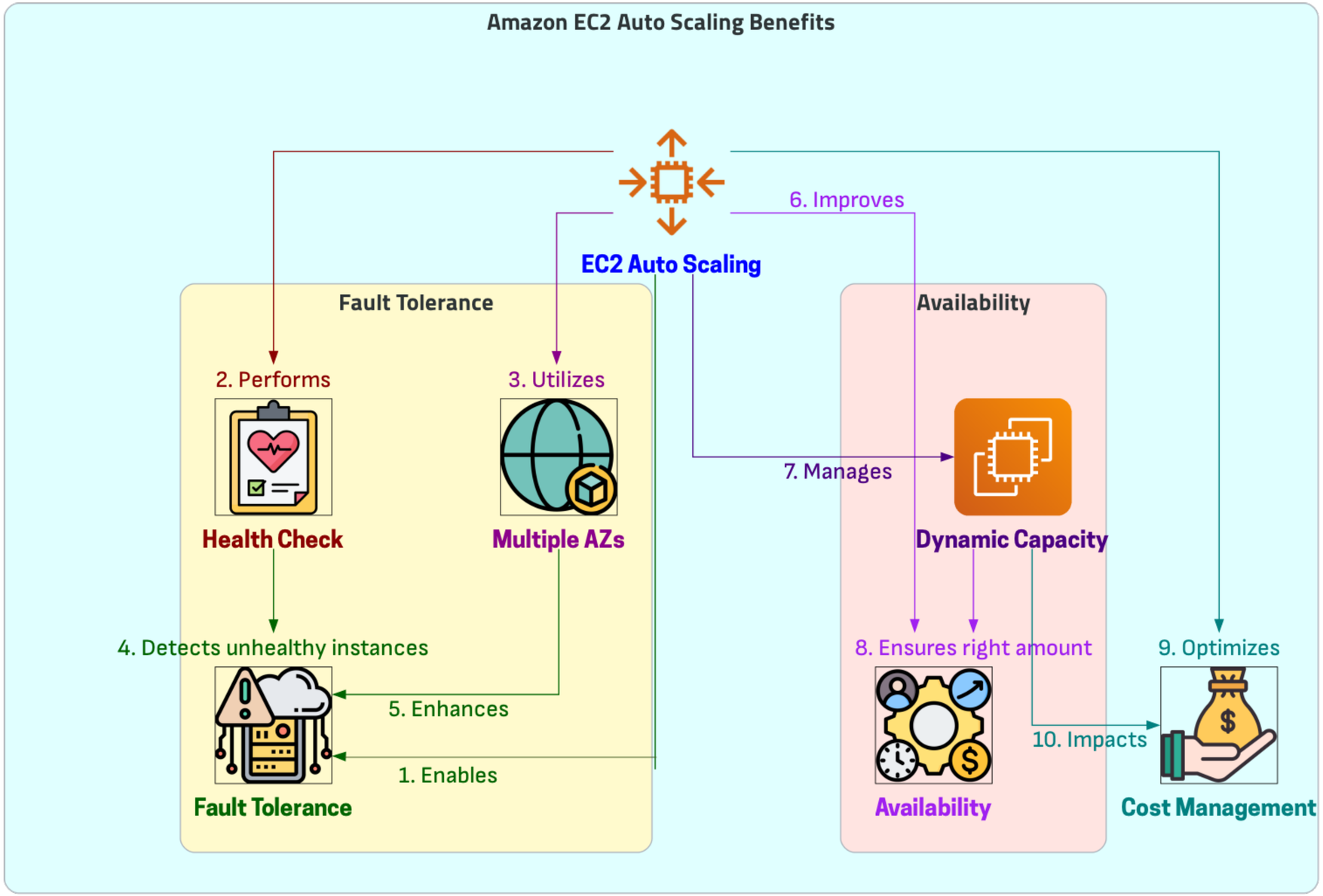


2️⃣. 📈 Better availability

⚖️ Ensures right amount of capacity

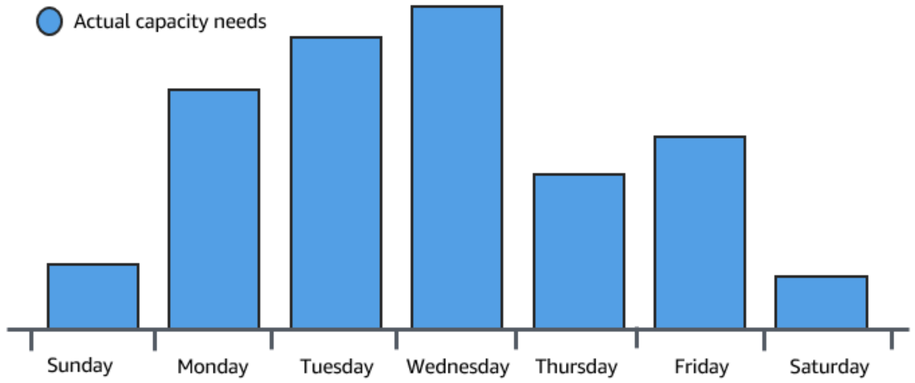
🚦 Handles current traffic demand

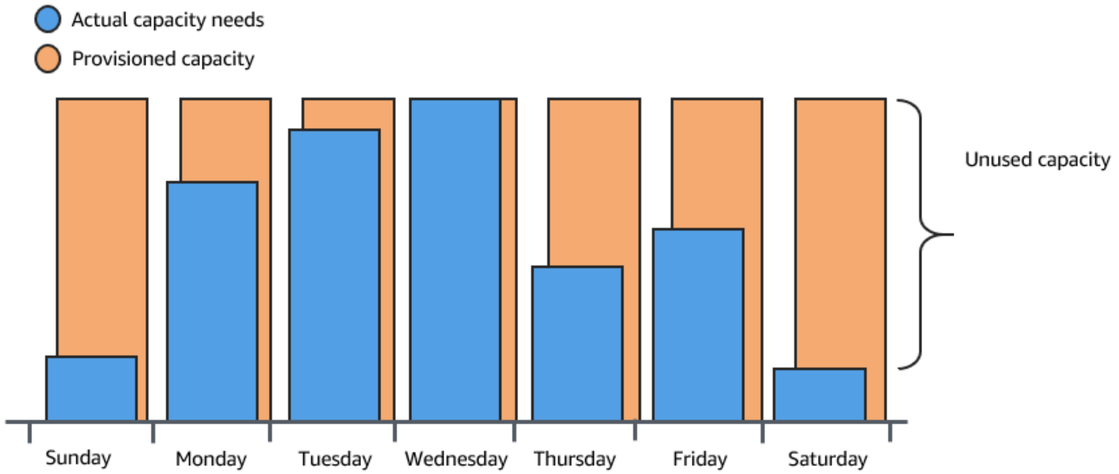
Amazon EC2 Auto Scaling Benefits

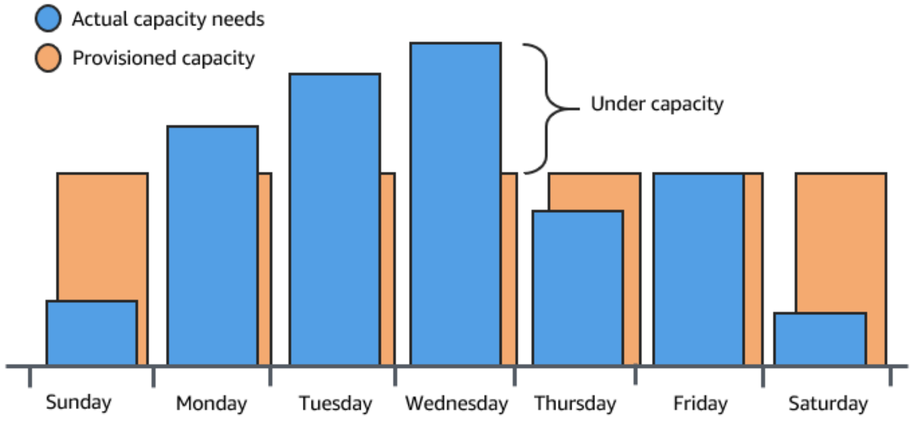


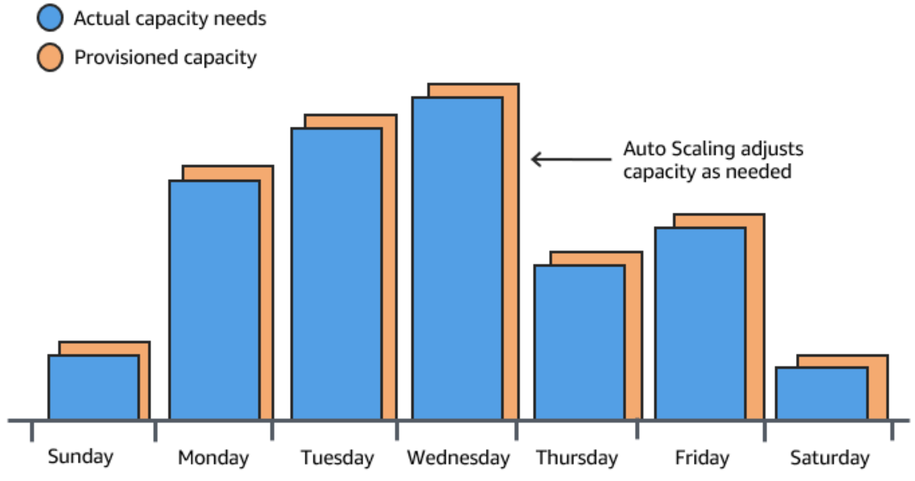
3️⃣. 💰 Better cost management	
📊 Dynamically adjusts capacity	🔼 Increases as needed
	🔽 Decreases as needed
💸 Pay for used instances only	🚀 Launches when needed
	🛑 Terminates when not needed

Cover Variable Demand







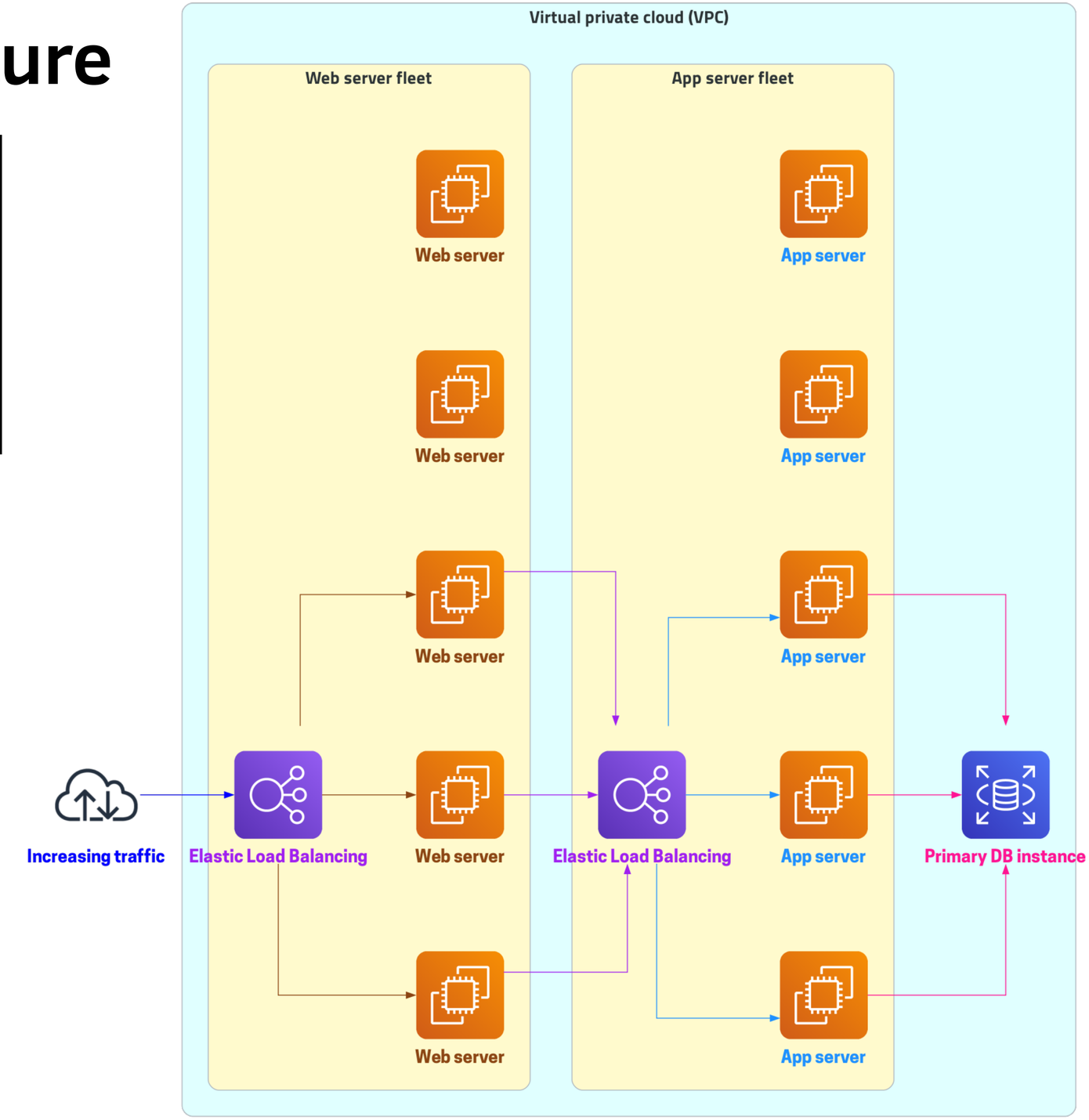


Typical Web app architecture

1️⃣. 🖥️ Multiple app copies
🔄 Simultaneously running
👥 Handles customer traffic volume

2️⃣. 🚀 EC2 instances
🔄 Identical cloud servers
📥 Process customer requests

3️⃣. 🤖 Amazon EC2 Auto Scaling	
▶️ Manages instance launch	
⏹️ Manages instance termination	
📊 Based on defined criteria	☁️ Amazon CloudWatch alarms

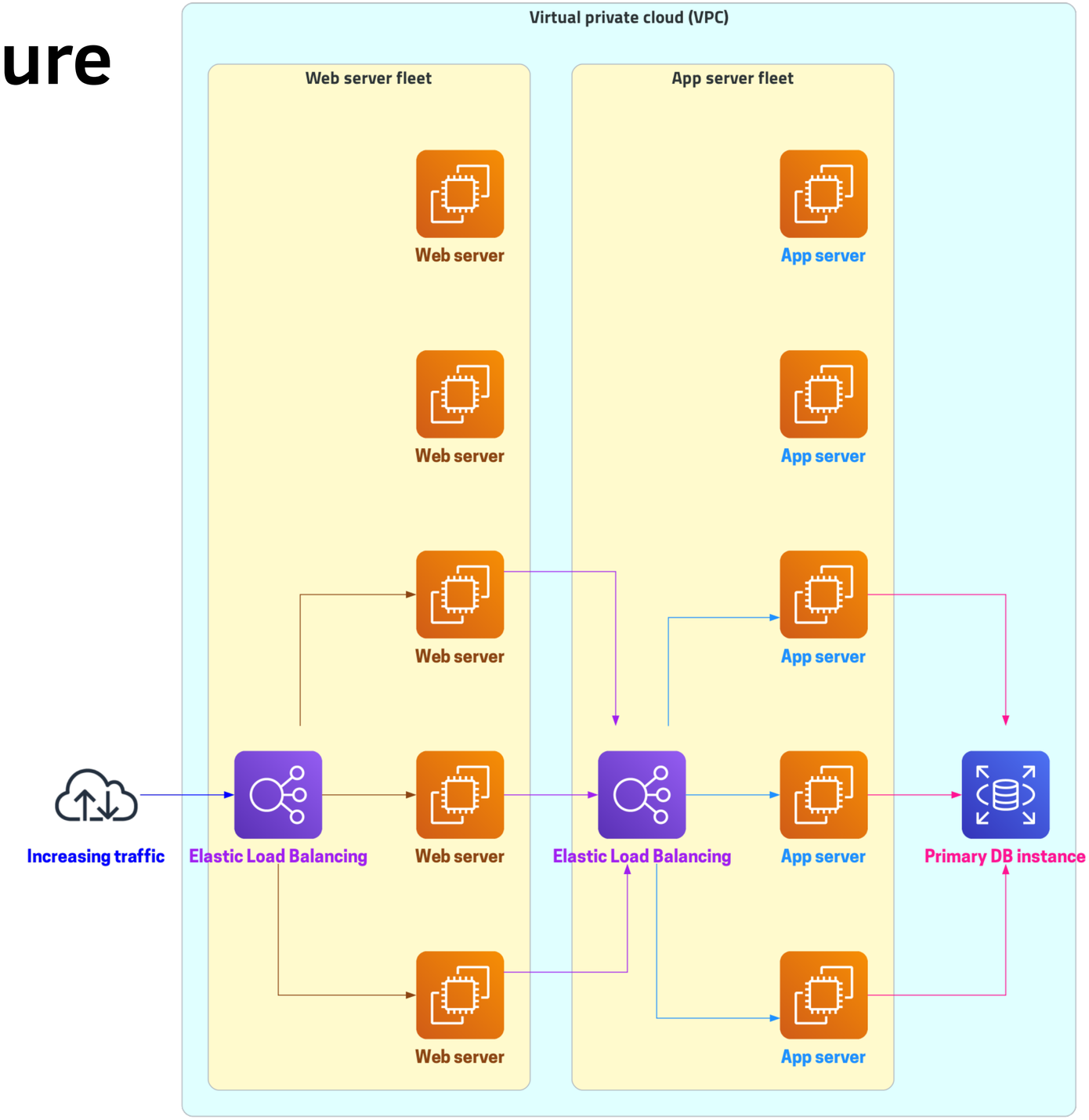


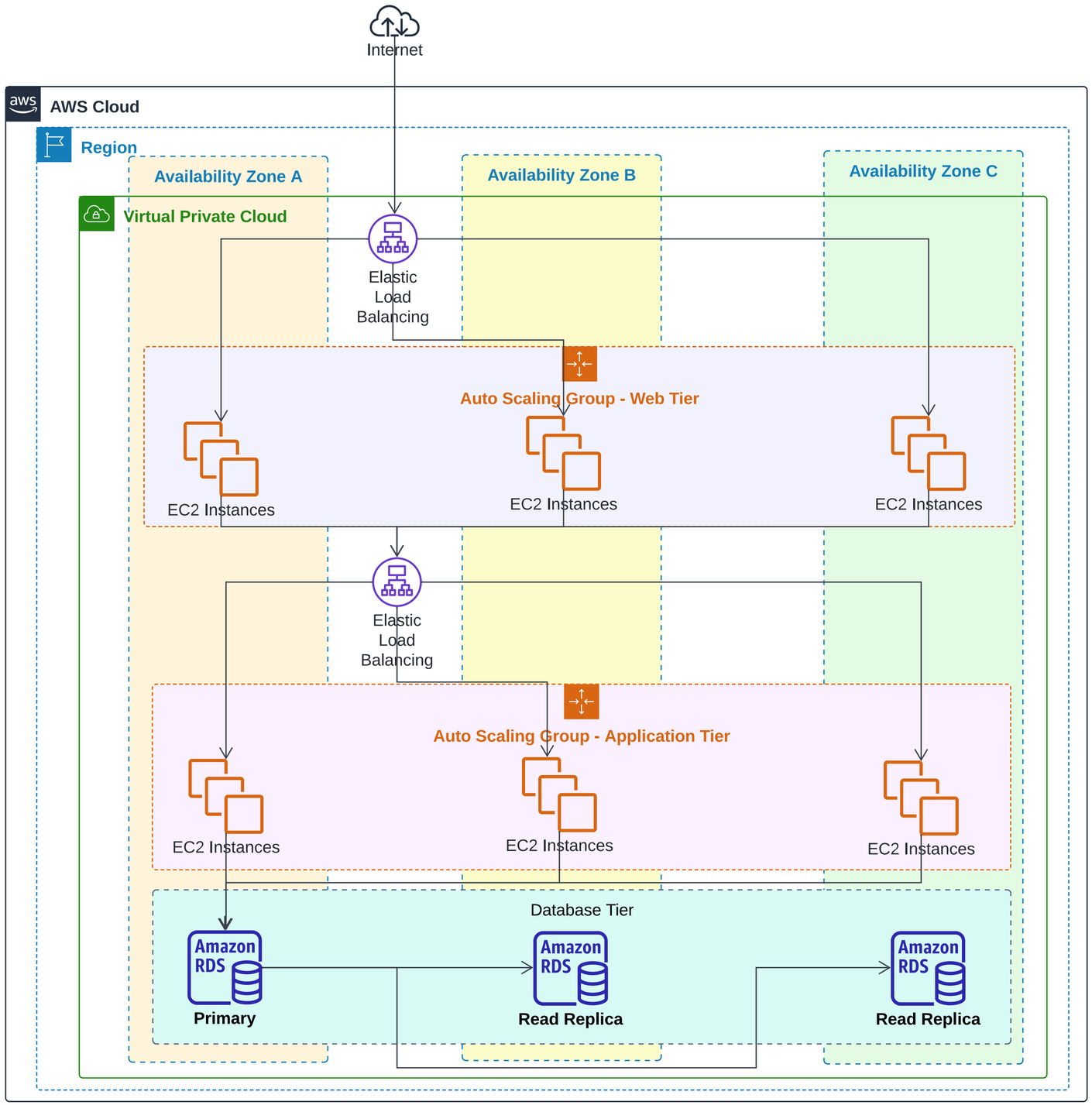
Typical Web app architecture

4. 🛡️ Benefits
🔼 Increased availability
🔁 Improved fault tolerance

5. 🧩 Flexibility
➕ Create multiple Auto Scaling groups
🏗️ Possible group per tier

6. ⚖️ Load balancer
🔀 Distributes traffic
🔗 Between instances in Auto Scaling groups

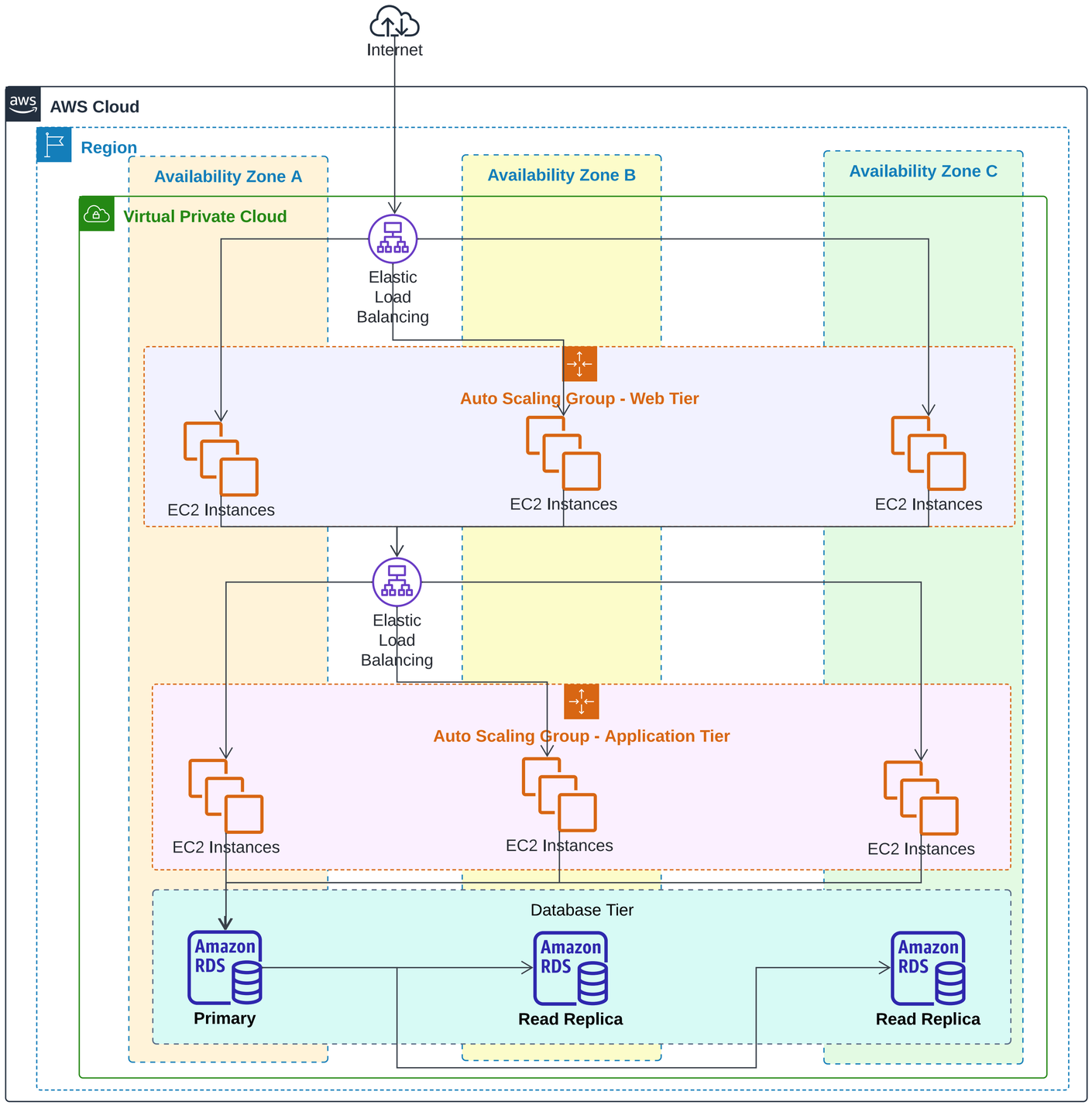




Distribute instances across Availability Zones

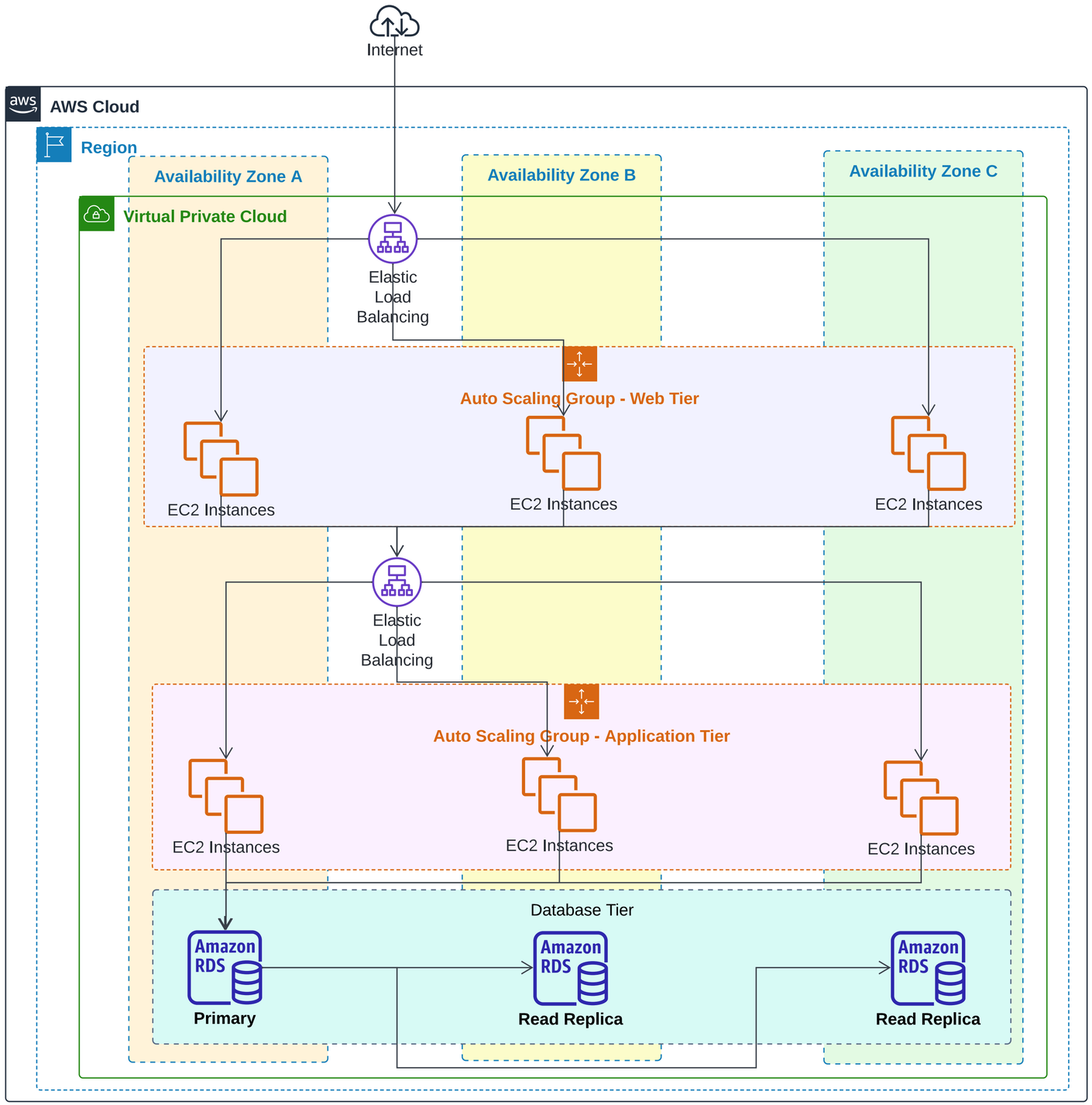
1️⃣. 🏙️ Availability Zones (AZs)
🗺️ Isolated locations in AWS Region
🔢 Multiple AZs per Region
🛡️ Designed for high availability
🆔 Identified by Region code + letter (e.g., us-east-1a)

2️⃣. 🌟 Benefits of using multiple AZs
📈 Increases application availability
🔒 Provides independence between zones



Distribute instances across Availability Zones

3️⃣. 🌐 VPC and Subnet Configuration	
🏗️ Create custom VPC and subnets	📍 Define subnets in each AZ
⚠️ Subnet restrictions	🔒 Must reside in one AZ
	🚫 Cannot span zones

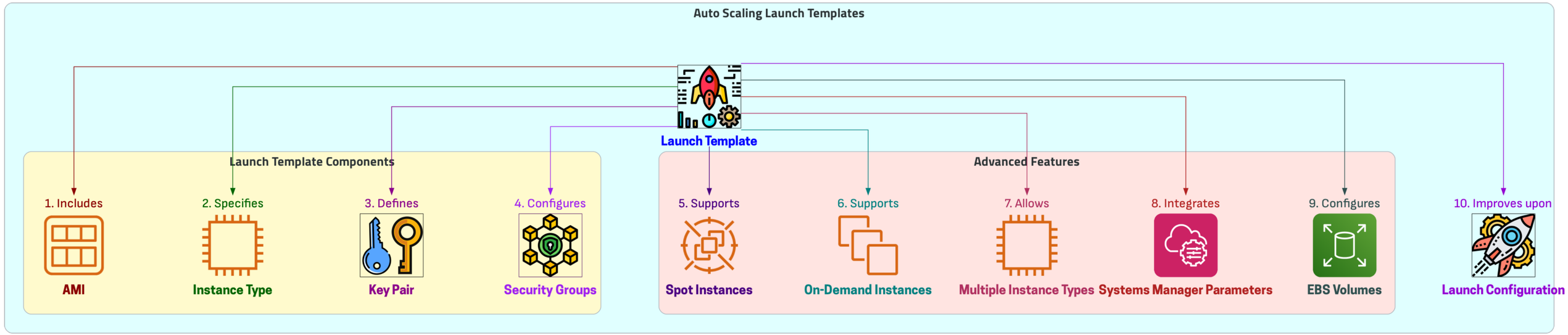


Distribute instances across Availability Zones

4️⃣. 🚀 Auto Scaling Group Deployment
🔧 Choose VPC and subnets for deployment
☁️ EC2 instances created in chosen subnets
🔗 Instances associated with specific AZ

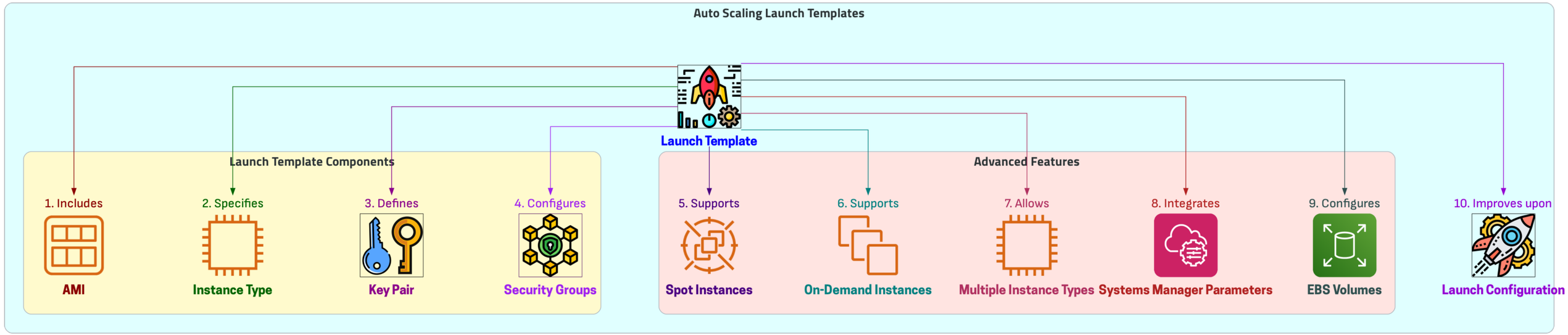
5️⃣. ⚖️ Instance Distribution	
🔄 Amazon EC2 Auto Scaling evenly distributes instances	
🎯 Goals	TOP High availability
	🛡️ Reliability

Auto Scaling launch templates



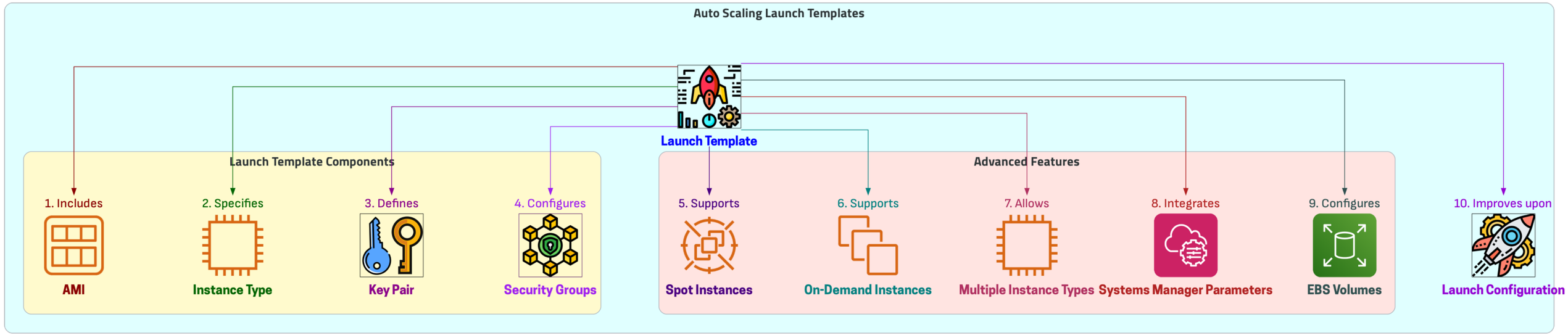
1️⃣. 📄 Launch Template Overview
🔧 Specifies instance configuration
🖼️ Includes AMI ID, instance type, key pair, security groups
📚 Supports multiple versions

Auto Scaling launch templates



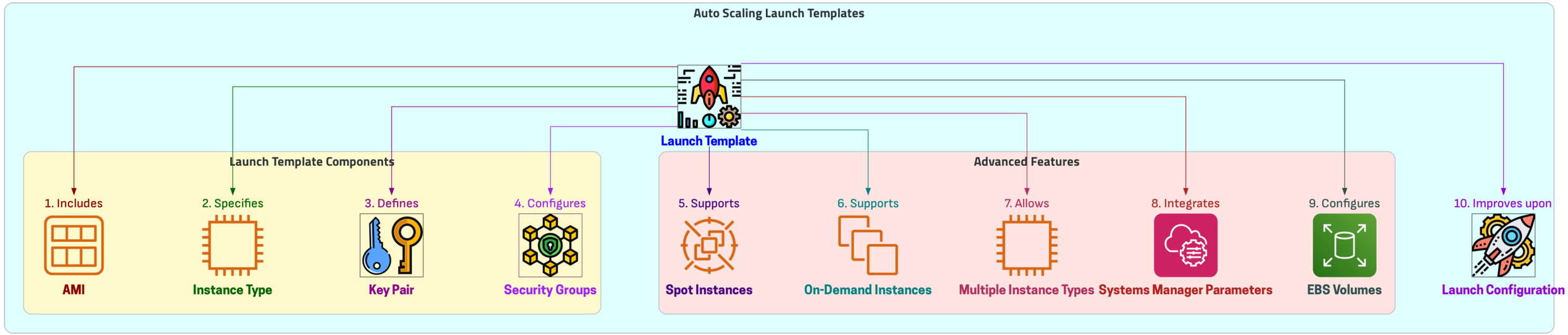
2️⃣. 🔢 Versioning Benefits	
🧩 Create subset of parameters	
♻️ Reuse for other versions	
🛠️ Example workflow	📋 Base configuration without AMI/user data
	🆕 New version with AMI/user data for testing
🗑️ Delete test versions when not needed	

Auto Scaling launch templates



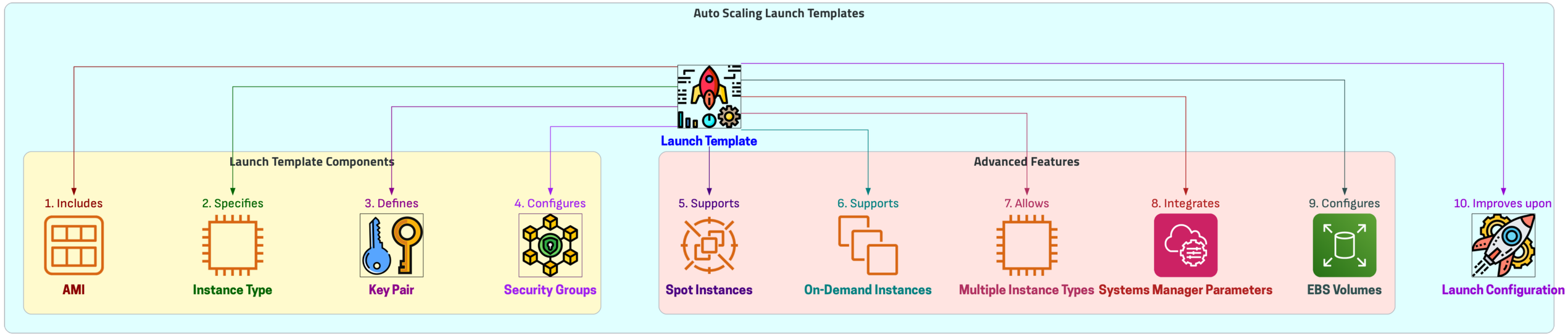
3️⃣. 🆚 Comparison with Launch Configurations	
⚠️ Some Auto Scaling features unavailable with launch configurations	🚫 Cannot launch both Spot and On-Demand Instances
	🚫 Cannot specify multiple instance types
✅ Launch templates required for these features	

Auto Scaling launch templates

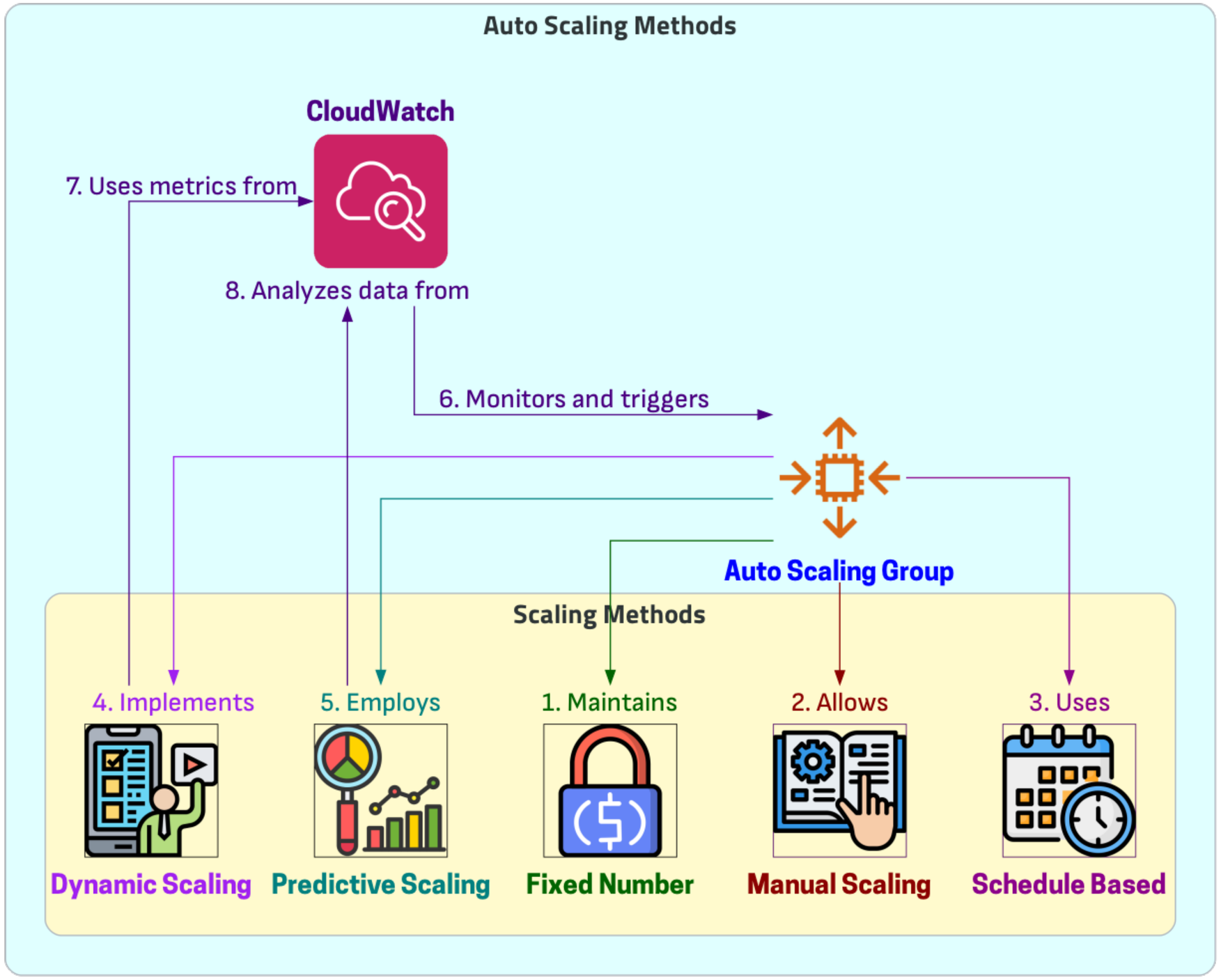


4️⃣. 🆕 Access to Newer EC2 Features
🔧 Systems Manager parameters (AMI ID)
💾 EBS Provisioned IOPS volumes (io2)
🏷️ EBS volume tagging
💪 T2 Unlimited instances
🔒 Capacity Reservations and Blocks
🏠 Dedicated Hosts

Auto Scaling launch templates



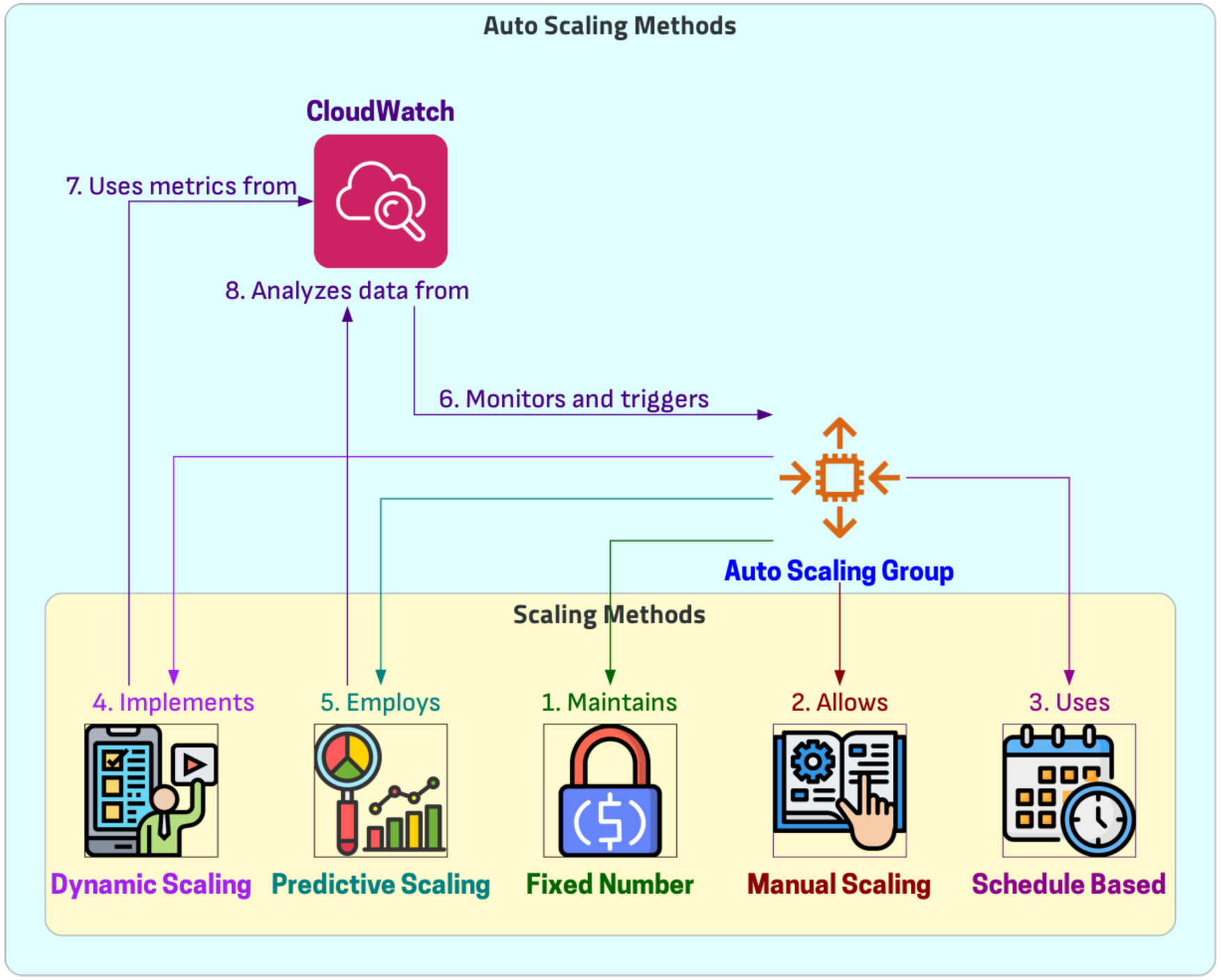
5️⃣. ⚙️ Launch Template Creation	
🔓 All parameters optional	
⚠️ Limitations	🖼️ Cannot add AMI later if not specified
	💻 Can add instance types later if AMI specified



Auto Scaling Methods

1️⃣. 🔢 Maintain a Fixed Number of Instances
🔁 Keep constant group size
🏥 Replace unhealthy instances
💯 Maintain desired capacity

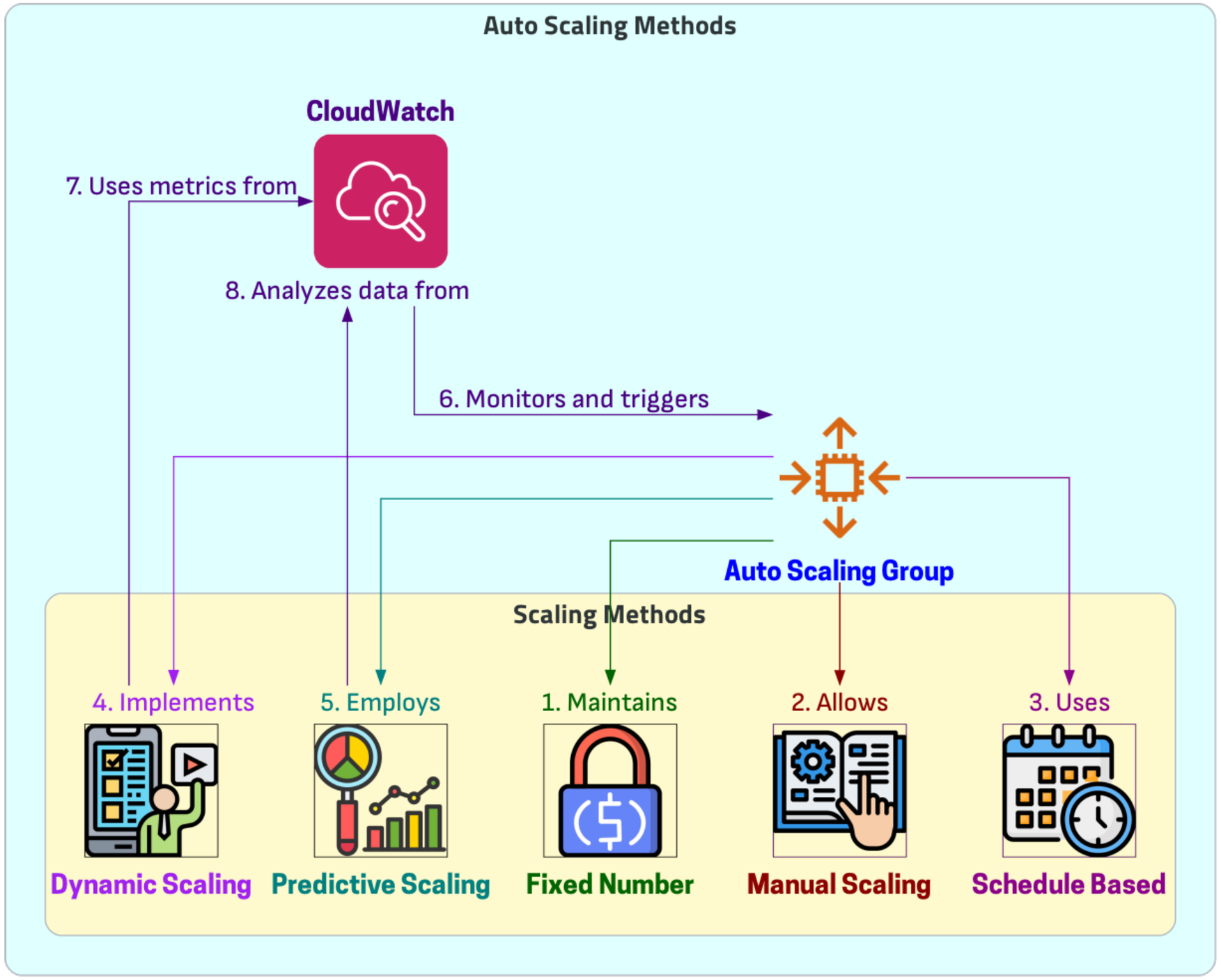
2️⃣. 🖐️ Manual Scaling
🎛️ Adjust desired capacity
🛑 Terminate specific instances
⚡ Immediate control



Auto Scaling Methods

3️⃣. 📅 Scale Based on a Schedule
🤖 Automate scaling actions
📊 Based on predictable demand
🚦 Align with expected traffic

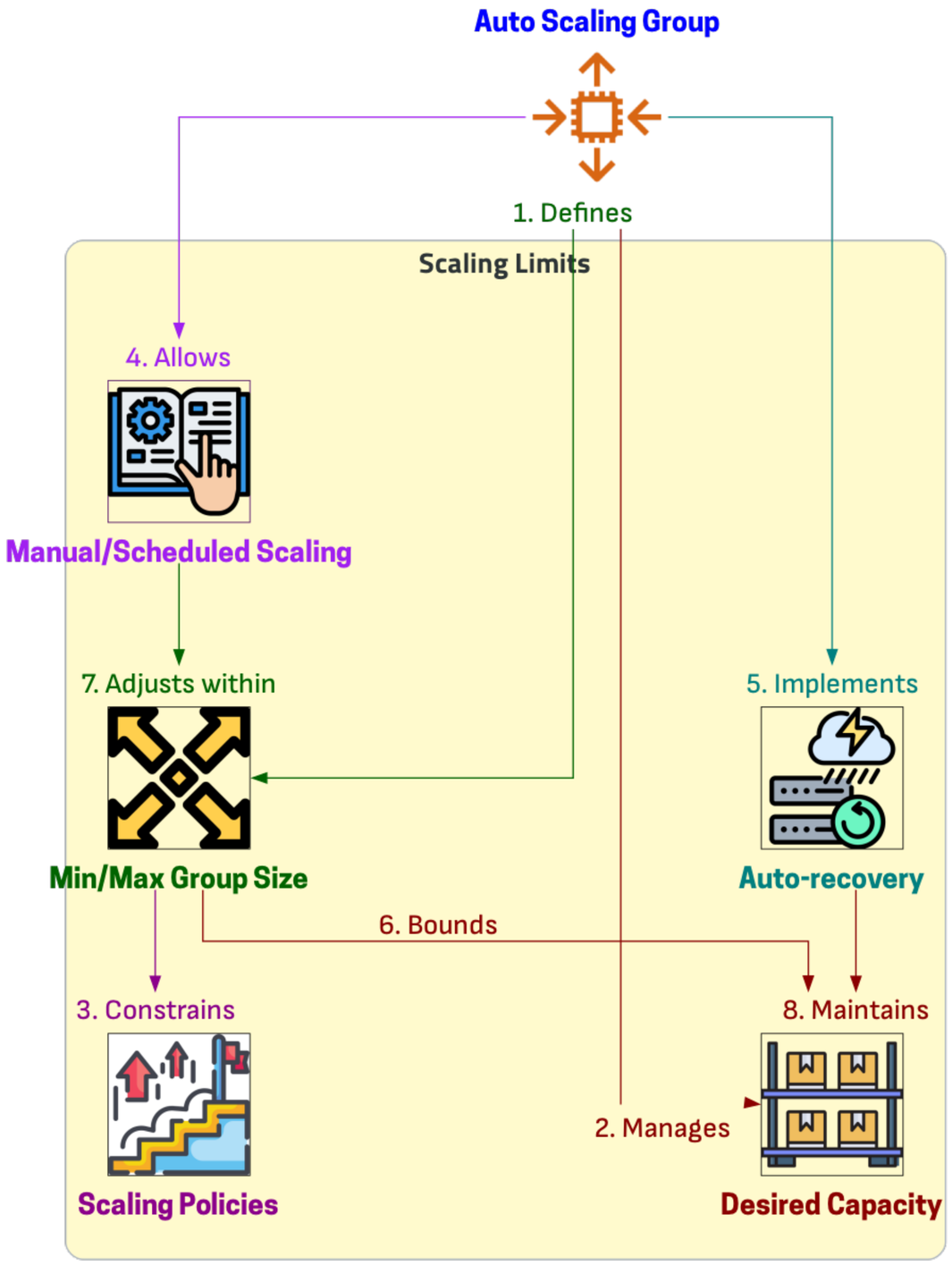
4️⃣. 📈 Scale Dynamically Based on Demand
⚖️ Use scaling policies
⏱️ Adjust capacity in real-time
🎯 Ensure optimal performance



Auto Scaling Methods

5️⃣. 🔮 Scale Proactively with Predictive Scaling
🤝 Combine predictive and dynamic scaling
🔄 Proactively adjust resources
📅 Ready for daily and weekly fluctuations

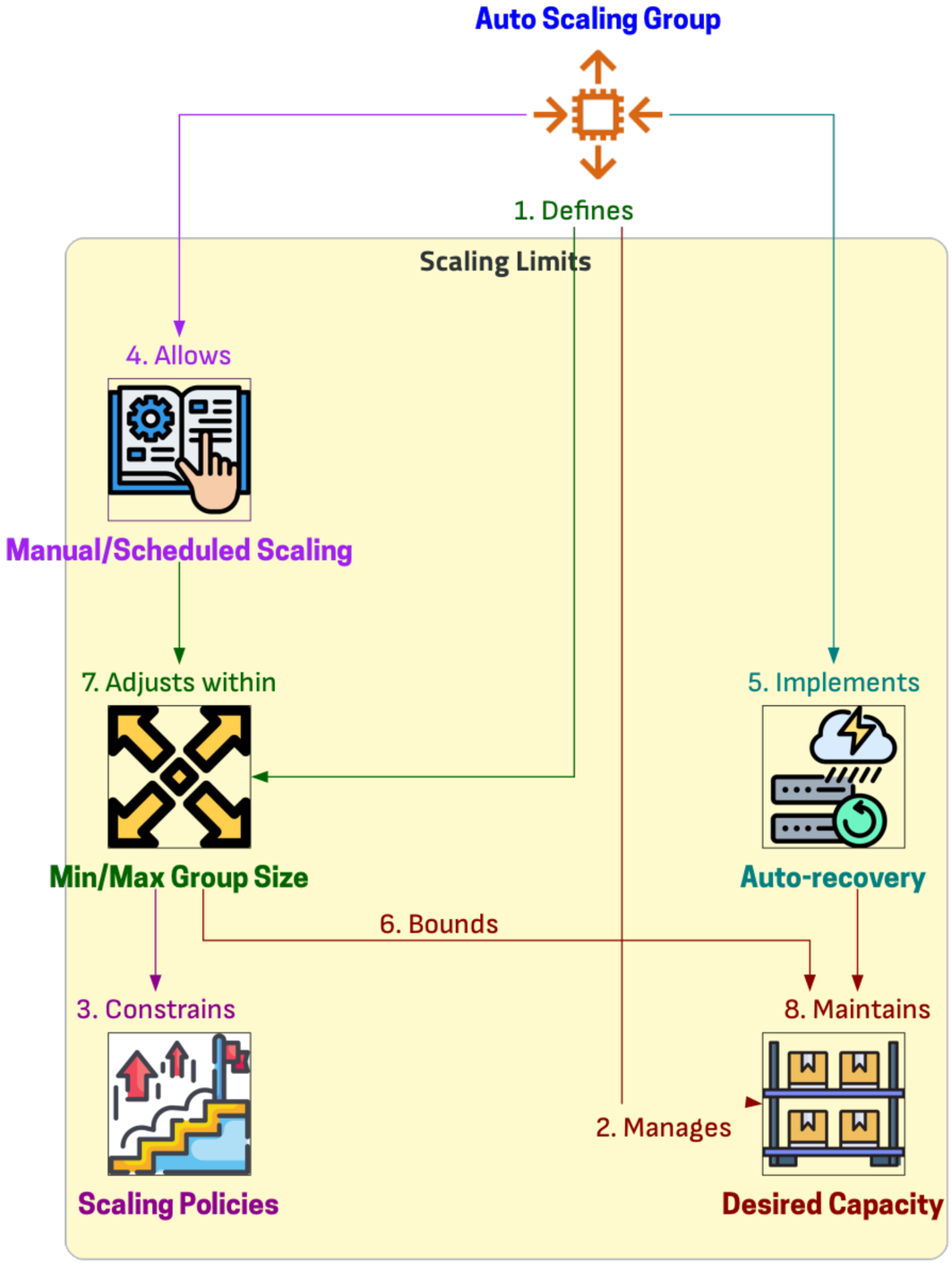
Set Scaling Limits for Your Auto Scaling Group



1️⃣. 🔭 Minimum and Maximum Group Size
🛡️ Defines boundaries for group size
🔀 Ensures operational flexibility
📏 Within predefined limits

2️⃣. 🎯 Desired Capacity Management
🏁 Initial group capacity at creation
⚖️ Maintains level unless altered
🔄 Affected by scaling policies or scheduled actions

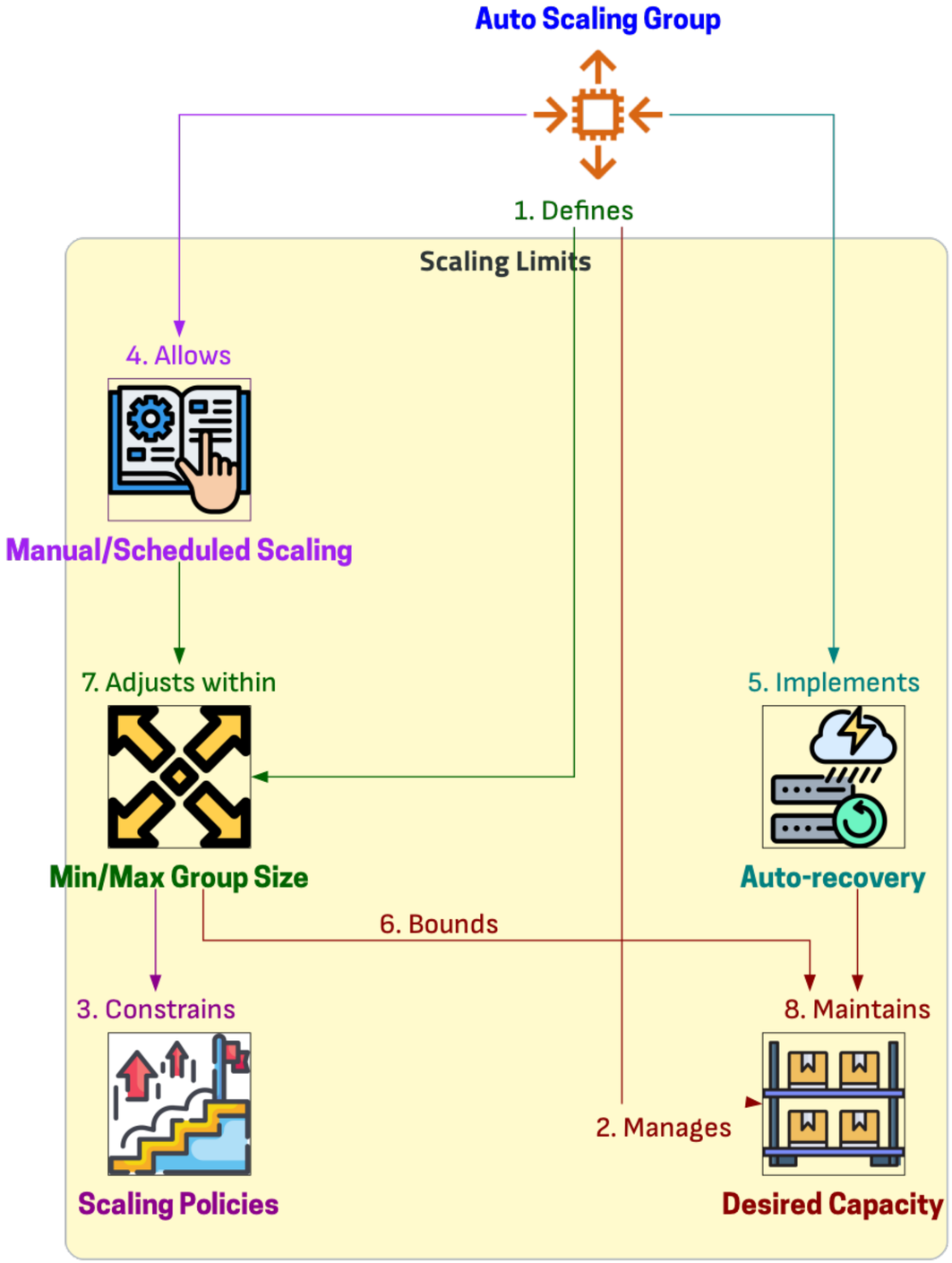
Set Scaling Limits for Your Auto Scaling Group



3️⃣. 🔽 Minimum Capacity Constraints
🛑 Prevents reduction below minimum
🛡️ Safeguards against resource inadequacy

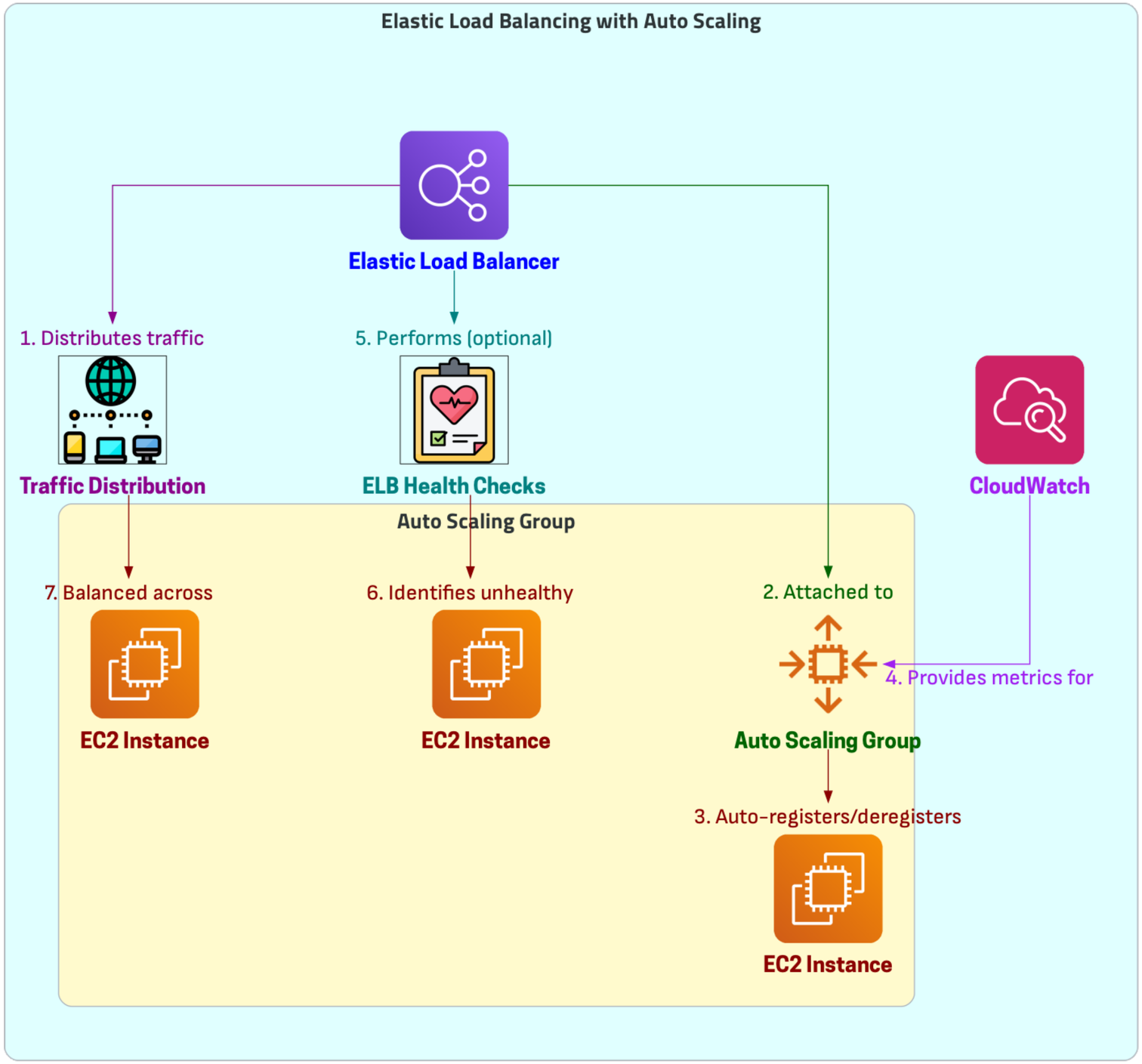
4️⃣. 🔼 Maximum Capacity Constraints
🚫 Prevents exceeding maximum size
🎛️ Controls resource overutilization

Set Scaling Limits for Your Auto Scaling Group



5️⃣. 🖐️ Manual and Scheduled Scaling Impact
🔧 Manual adjustment capability
⏰ Scheduled actions for capacity changes
📏 Within set minimum and maximum limits

6️⃣. 🔄 Auto-recovery of Desired Capacity
🚀 Automatically launches new instances
🎯 Maintains desired capacity
🛡️ Handles unexpected terminations
💪 Ensures resilience and continuity

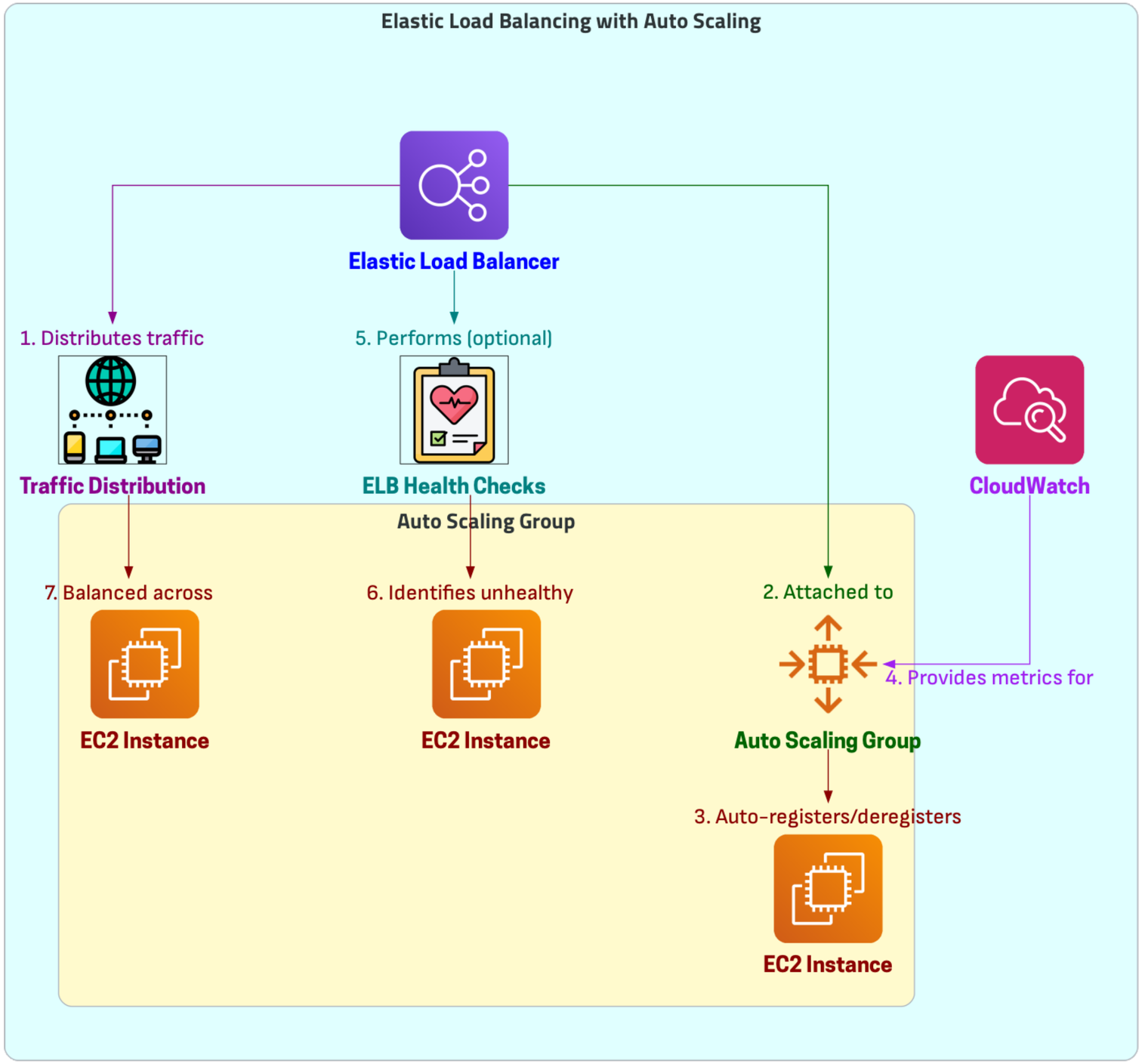


Elastic Load Balancing with Auto Scaling

1️⃣. 🔄 Automatic Traffic Distribution
⚖️ Evenly distributes incoming traffic
🛡️ Prevents instance overload

2️⃣. 🛠️ Attach Load Balancer to Auto Scaling Group
🎯 Single point of contact for traffic
🔧 Simplifies setup
📈 Enhances efficiency

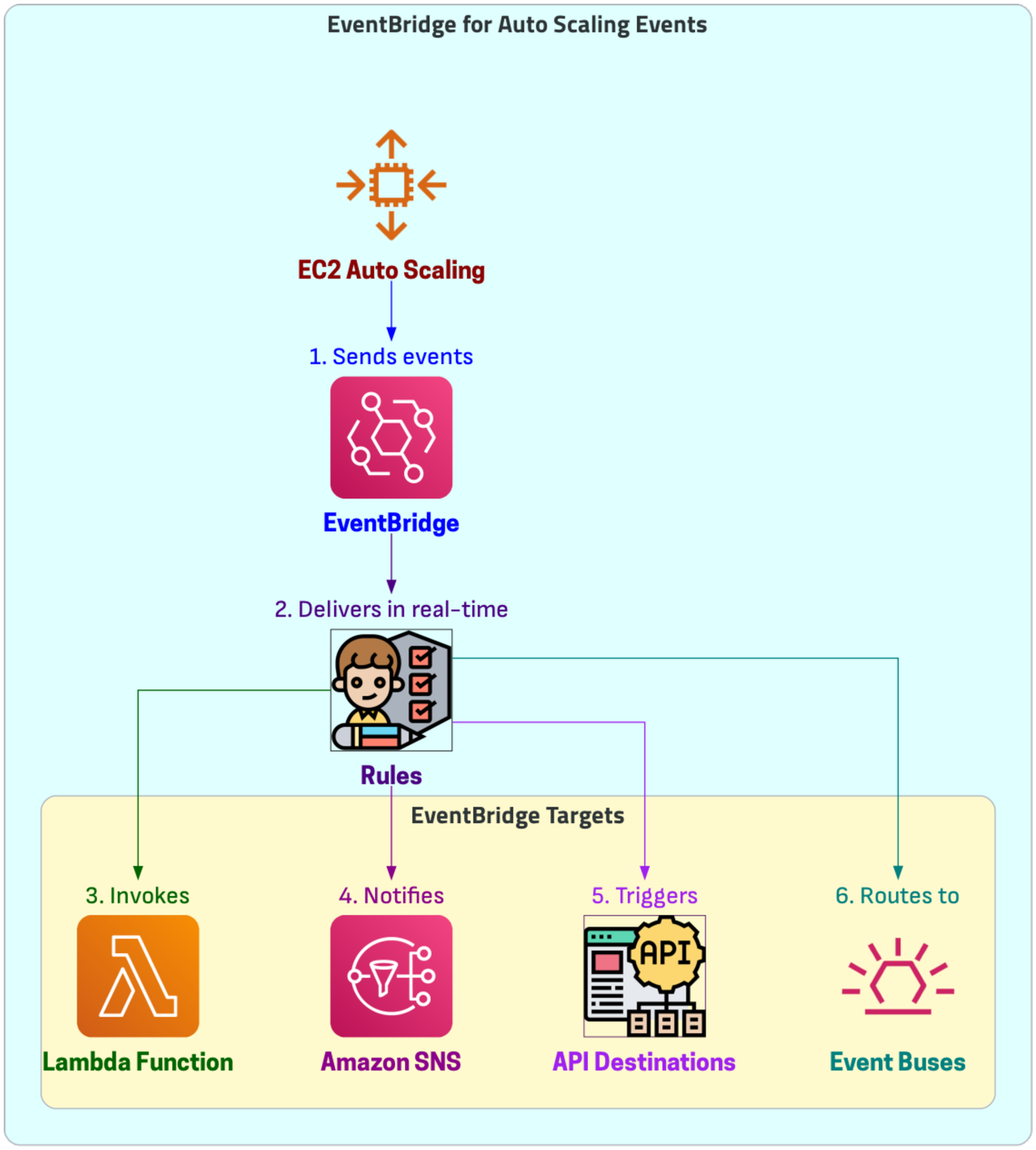
3️⃣. 🚫 No Manual Instance Registration Needed
🔌 Automatic instance registration
🔥 Automatic instance deregistration



Elastic Load Balancing with Auto Scaling

4️⃣. 📈 Use ELB Metrics for Scaling
📊 Dynamic scaling based on demand
🔢 Utilizes ELB metrics
📩 Application Load Balancer request count

5️⃣. ✅ Optional: ELB Health Checks
🩺 Identifies unhealthy instances
🔄 Replaces unhealthy instances
💪 Enhances reliability



Use EventBridge for Auto Scaling Events

1️⃣. 🔍 EventBridge Introduction
👀 Monitors resources
🎬 Initiates actions
📏 Based on rules

2️⃣. ⚡ Real-time Event Delivery
🚀 Nearly instantaneous delivery
⏱️ Allows timely responses

3️⃣. 📜 Rules for Programmatic Actions	
🛠️ Create rules in EventBridge	
🔔 Invoke actions or notifications	
🖥️ Respond to EC2 Auto Scaling events	▶️ Launching instances
	⏹️ Terminating instances

4️⃣. 🎯 Wide Range of Targets
λ AWS Lambda functions
📣 Amazon SNS topics
🔌 API destinations
🚌 Event buses
🔀 Diverse set of responses

Thanks for Watching