---

# 🔹 Configure Scaling for an Azure App Service Plan

Azure App Service provides two main scaling options:

- **Scale Up (Vertical Scaling)** – Increase instance size (CPU, RAM, storage).
- **Scale Out (Horizontal Scaling)** – Increase the number of instances running your app.

---

## Step 1: Navigate to App Service Plan

1. Log in to Azure Portal.
2. In the search bar, type **App Service Plans** → Select your plan.
   *(Or go to your App Service → Under "Settings", click **Scale up / Scale out**.)*

---

## Step 2: Scale Up (Vertical Scaling)

Use this when you want **more powerful compute resources**.

1. In the left-hand menu, select **Scale up (App Service plan)**.
2. Choose a higher pricing tier (e.g., from **B1 Basic** → **S1 Standard** or **P1v3 Premium**).
   - Higher tiers unlock features like **staging slots, autoscaling, VNET integration, backups**.
3. Click **Apply**.

---

# Step 3: Scale Out (Horizontal Scaling)

Use this when you want to **run multiple instances** of your app.

## Option A: Manual Scale Out

1. Go to **Scale out (App Service plan)**.
2. Choose **Manual scale**.
3. Select the **Number of instances** (e.g., 1 → 3).
4. Click **Save**.
   👉 Azure will distribute incoming traffic across all instances automatically (via Load Balancer).

## Option B: Autoscale (Recommended for production)

1. Go to **Scale out (App Service plan)**.
2. Select **Custom autoscale**.
3. Configure settings:
   - **Autoscale condition**: Add a rule.
   - Example:
     - If **CPU Percentage > 70%** for **10 minutes**, increase instances by 1.
     - If **CPU Percentage < 30%** for **10 minutes**, decrease instances by 1.
   - **Instance limits**: Set minimum (e.g., 1), maximum (e.g., 5), and default instance count.
   - (Optional) **Schedule-based rules**: Scale differently during business hours vs. nights/weekends.
4. Click **Save**.

---

## Step 4: Verify Scaling

1. Open your App Service → **Overview**.
2. Under **Essentials**, check the number of instances running.
3. You can test scaling by simulating load (using tools like Apache JMeter or Azure Load Testing).

---

## Best Practices

- Always set **min/max limits** to control costs.
- Use **autoscale with metrics** like CPU, Memory, or HTTP Queue Length for production workloads.
- Use **scale-up + scale-out together** for maximum flexibility.
- Monitor with **Application Insights** to fine-tune scaling rules.

---

✅ **Summary:**

- **Scale Up** → Upgrade to stronger compute resources.
- **Scale Out** → Add multiple instances (manual or autoscale).
- **Autoscale** → Automatically adjust based on workload and schedule.

---