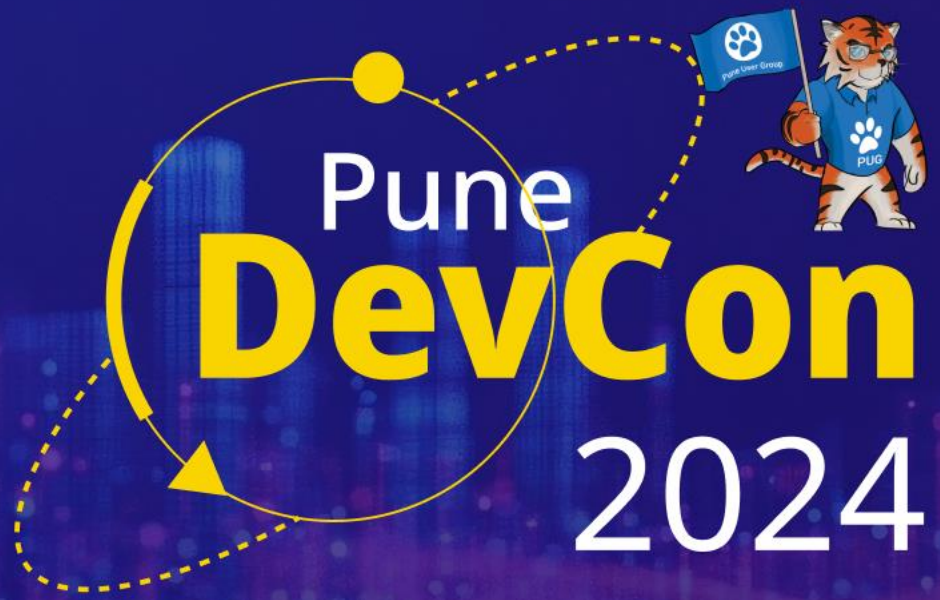
PUG
Pune User Group

#PuneDevCon



Designing “Orchestrable” Applications

Raj Chaudhuri
Rajware Services Pvt. Ltd.

Pune DevCon 2024

Thank you!

PUG Pune User Group

to Our Sponsors

Icertis

RAPID CIRCLE

SYNERGETICS
Educate | Advise | Implement | Managed

QuestPond

NSquare

SGM EDUCATION GROUP

/PuneUserGroup | /PuneUserGroup | /PuneUserGroup

Agenda

- What are “Orchestrable” Applications?
- Seven Steps
- Best practices for [platform] developers:
  - Today, [platform] is .NET Core

What are “Orchestrable” applications?

What is Orchestration?

Automated management of applications

Initial deployment, configuration, monitoring, error recovery, scaling and updating

Resource co-ordination

Optimal allocation and coordination of resources like CPU, memory, storage, and networking

Centralized control

A unified platform to manage diverse workloads and their dependencies across distributed systems

Minimized Human Intervention

Near-zero manual management

What are “Orchestrable” applications?

Applications that allow Orchestration systems to:

- Deploy them anywhere appropriate as needed
- Provide them with required services, such as storage and networking
- Configure them, sometimes with sensitive information
- Start and initialize them
- Monitor their health and performance
- Restart them on critical failure
- Let them handle non-critical failure themselves
- Scale them as needed, both up and down
- Update them safely while maintaining availability and integrity

Seven Steps

Seven Steps to “Orchestrable” Applications

- Containerize your application
- Provide health checks
- Specify application requirements
- Design for scalability
- Generate metrics
- Enable automated updates
- Ensure confidentiality

Containerize your Application - Why

- Consistency

  Container images contain the application and all dependencies, and are immutable

- Speed of start and restart

  Therefore, creating a new container (an instance of the application) is cheap and fast

- Isolation

  Each container has its own (controlled) private process space, storage and networking

All of which make them fast and safe for automated deployment

Containerize your Application - How

- One top-level process, responsible for keeping the container alive, logging and graceful shutdown
- Log to standard output, with configurable log levels
- Keep data separate from code. That way, data can be stored on mounted external storage (volumes)
- Be fully configurable via environment variables or configuration files
- Start fast, fail when needed.

Provide Health Checks - Why

- Automated systems can detect (and correct) crashes
- They cannot detect “hanging” or application errors

Provide Health Checks - How

- Crash for problems where administrator intervention is required
- Do not crash for user-level problems
- Provide a way for automated checking

Provide Health Checks - Reference

- Health checks in ASP.NET Core | Microsoft Learn

Specify application requirements -Why

- Prevents over/under-provisioning of CPU, memory, and storage
- Ensures services and dependencies are colocated or scheduled correctly
- Helps orchestrators provide the optimal environment for the application

Specify application requirements - How

- Specify resource needs (CPU, memory) per application component
- Specify storage needs, also per component, unifying when needed
- Specify networking requirements such as ports and policy
- Specify affinity rules to colocate or contralocate dependencies

Design for scalability - Why

- Supports increased traffic without performance degradation
- Enables the application to handle failures by redistributing the load
- Allows dynamic scaling to save costs during low traffic

Design for scalability - How

- Favor stateless design
- Use asynchronous programming for better responsiveness
- Design for load balancing to handle increased traffic

Design for scalability - Reference

- Configure ASP.NET Core to work with proxy servers and load balancers | Microsoft Learn
- Configure ASP.NET Core Data Protection

Generate Metrics - Why

- Helps monitor application performance and health
- Enables orchestration systems to make scaling decisions
- Simplifies debugging and identifying bottlenecks
- Tracks performance for SLAs and audits

Generate Metrics - How

- Log health metrics for orchestration systems
- Use middleware to measure request times and errors
- Define meaningful custom metrics (e.g., user logins)
- Expose Prometheus-compatible metrics (prometheus-net)

Enable Automated Updates - Why

- Ensures updates are rolled out seamlessly with minimal impact
- Quickly applies patches and mitigates vulnerabilities
- Deploys new features rapidly *without manual intervention*
- Prevents discrepancies between environments (e.g. test and production)

Enable Automated Updates - How

- Decide update strategy for each component (e.g. blue-green or rolling deployments)
- Automate database migrations (e.g., EF Core)
- Version APIs for compatibility during upgrades
- Implement feature flags for dynamic changes

Ensure Confidentiality - Why

- In automated systems, maintenance of data and especially configuration is also automated
- Although isolation is already provided, we do not want even accidental leaks or unauthorized access
- Therefore, we need to protect these things for privacy and compliance

Ensure Confidentiality - How

- Secure sensitive configuration with minimal exposure
- Encrypt sensitive data
- Integrate with secure secret managers (like Azure Key Vault or Hashicorp Vault)

In Conclusion

- Design robust, scalable, and secure applications by keeping orchestration in mind.
- Tailor practices to your orchestration platform.
- Automation, monitoring, and security are important for *all* applications
  - No matter big or small
  - And not just for “microservices”

Pune DevCon 2024

Thank you!

PUG Pune User Group

to Our Sponsors

Icertis

RAPID CIRCLE

SYNERGETICS
Educate | Advise | Implement | Managed

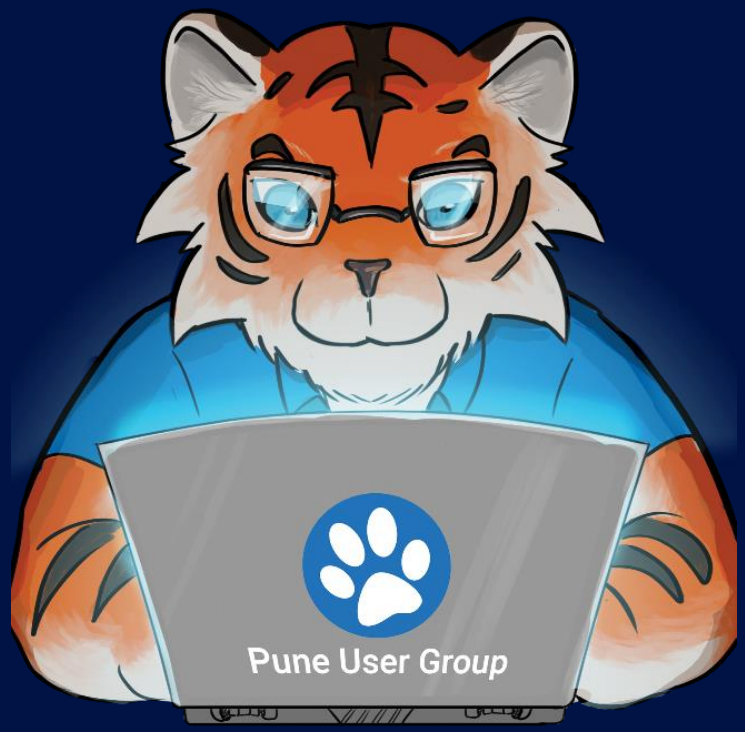
QuestPond

NSquare

SGM EDUCATION GROUP

/PuneUserGroup | /PuneUserGroup | /PuneUserGroup

Pune DevCon 2024
Proudly hosted by
Pune User Group



Pune DevCon 2024