

Introdução à Microsoft Azure

Computação em Nuvem

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

Traditional Datacenter



Microsoft Azure



Virtual Machines



Cloud Services



Web Sites



Mobile Services



Applications

Applications

Applications

Web Applications

Mobile Applications

Data

Data

Data

Data

Data

Firewall

Firewall Rules

Firewall Rules

Network

Virtual Network

Virtual Network

O/S

O/S

Virtualization

Hardware



Visão Geral da Microsoft Azure

- **Definição:** Plataforma de nuvem projetada para simplificar a construção de aplicativos modernos.
- **Benefícios Principais:**
 - Ajuda a criar aplicativos **escalonáveis, confiáveis e de fácil manutenção**.
 - Suporta **hospedagem total** ou **extensão** de aplicativos locais existentes.
- **Suporte e Produtividade:**
 - Suporte a linguagens de programação populares: Python, JavaScript, Java, .NET, Go.
 - Biblioteca abrangente do SDK para linguagens como .NET, Go, Java, JavaScript, Python.
 - Amplo suporte em ferramentas conhecidas: VS Code, Visual Studio, IntelliJ e Eclipse.

Cenários Comuns de Aplicação no Azure

- **Hospedagem de Aplicações na Nuvem:**
 - Hoster a pilha completa: Aplicativos Web, APIs, bancos de dados e serviços de armazenamento.
 - Aproveita a escalabilidade, alta disponibilidade e segurança nativas do Azure.
- **Arquiteturas Baseadas em Contêiner:**
 - Modernizar aplicativos através da conteinerização.
 - Soluções para gerenciamento de contêineres em escala (microserviços).
- **Arquiteturas Modernas Sem Servidor (**Serverless**):**
 - Usar o **Azure Functions** para soluções orientadas a eventos (HTTP, uploads de arquivos, eventos de fila).
 - Permite escrever apenas o código necessário, abstraindo servidores e código de estrutura.

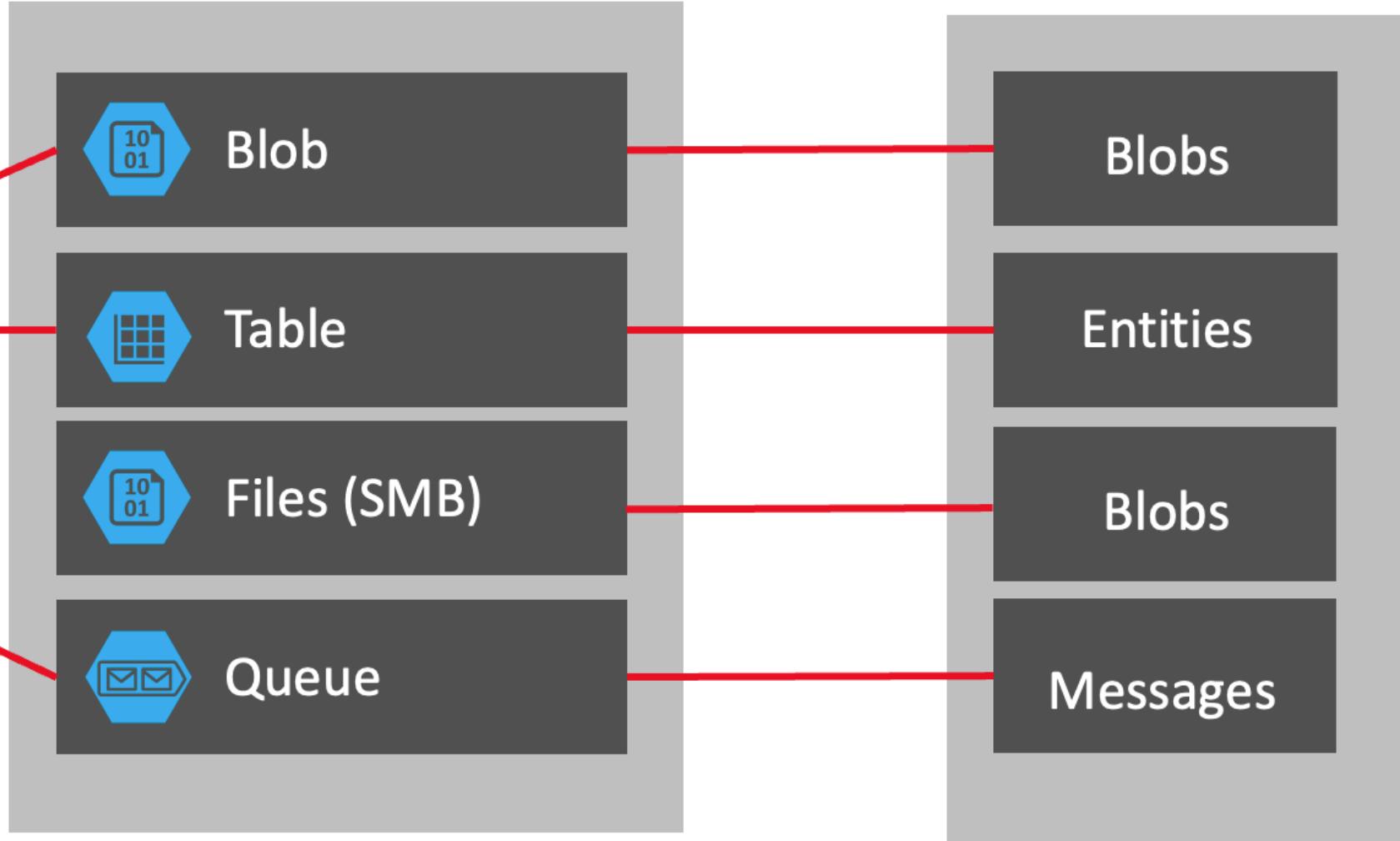
Principais Serviços I: Computação e Hospedagem

Serviço	Descrição	Modelo
Serviço de Aplicativo do Azure	Hospedagem de serviço completo para aplicativos Web e APIs (.NET, Java, Python, etc.). Azure gerencia infraestrutura (escala, HA).	PaaS (Plataforma como Serviço)
Azure Functions	Plataforma de computação sem servidor para pequenos códigos acionados por eventos. Ideal para APIs sem servidor ou orquestração de eventos.	Serverless
Máquinas Virtuais do Azure (VMs)	Oferece mais controle sobre o ambiente de computação (Linux e Windows).	IaaS (Infraestrutura como Serviço)
Aplicativos de Contêiner do Azure	Executar aplicativos em contêineres em uma plataforma sem servidor sem gerenciar a orquestração ou infraestrutura.	Serverless Contêiner

Principais Serviços II: Dados, Armazenamento e Messaging

- **Dados (Relacional e NoSQL):**
 - **Banco de Dados SQL do Azure:** Versão totalmente gerenciada baseada em nuvem do SQL Server.
 - **Azure Cosmos DB:** Banco de dados NoSQL totalmente gerenciado, oferecendo múltiplas APIs (incluindo MongoDB, Cassandra).
 - **Banco de Dados do Azure para PostgreSQL/MySQL/MariaDB:** Serviços de banco de dados open-source totalmente gerenciados.
- **Armazenamento de Blobs do Azure:** Permite armazenar e recuperar grandes quantidades de arquivos na nuvem (altamente escalável, alta disponibilidade).
- **Messaging (Filas e Eventos):** Broker de mensagens corporativo gerenciado, ideal para desacoplar aplicativos (integrações P2P e publicar-assinar).

Azure Storage Services



`http://<storage acct>.blob.core.windows.net/<container>/<blob>`
`http://<storage acct>.table.core.windows.net/<table>`
`http://<storage acct>.queue.core.windows.net/<queue>`

Conceitos Chave de Arquitetura I: Regiões e Resiliência

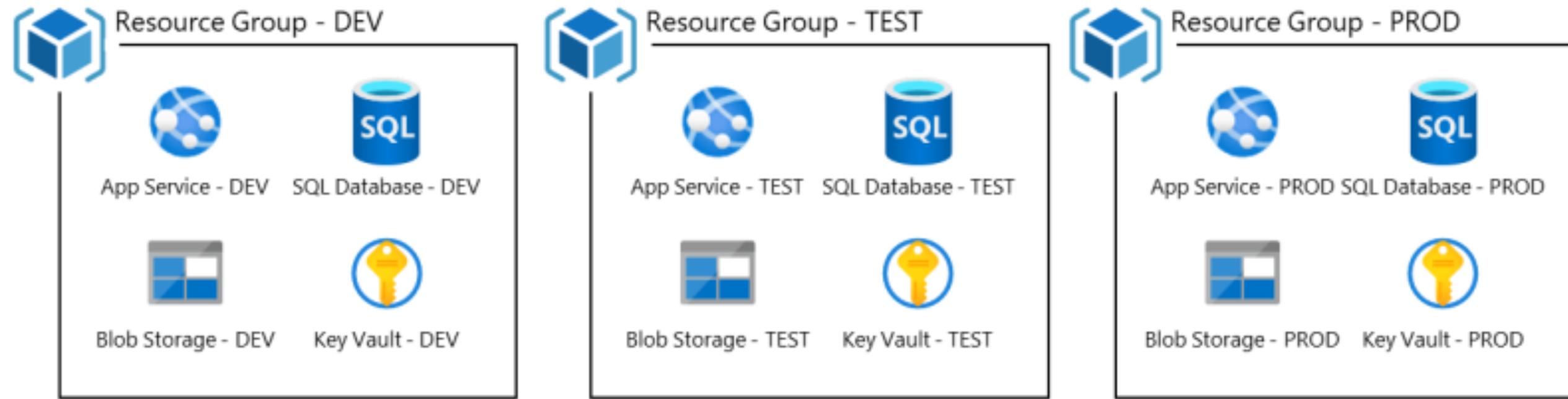
- **Regiões do Azure:**
 - Um conjunto de **datacenters** implantados dentro de um perímetro definido por latência.
 - Conectados por uma rede regional dedicada de baixa latência.
 - Permitem fornecer resiliência entre regiões.
- **Regras de Latência:**
 - É recomendável manter todos os recursos de uma solução na **mesma região** para minimizar a latência entre componentes.
 - **Exemplo:** Serviço de Aplicativo, Banco de Dados e Armazenamento de Blobs devem estar na mesma região.

Conceitos Chave de Arquitetura I: Regiões e Resiliência

- **Grupo de Recursos do Azure (Resource Group):**
 - Um **contêiner lógico** que agrupa recursos do Azure. Cada recurso pertence a um grupo de recursos.
 - Normalmente, agrupa todos os recursos necessários para uma solução (App Service, DB, Key Vault).
 - A criação do Grupo de Recursos é frequentemente o primeiro passo na implantação de um aplicativo.

Conceitos Chave de Arquitetura II: Ambientes e DevOps

- **Ambientes de Implantação:** Para criar ambientes separados (Desenvolvimento, Teste, Produção), crie um **conjunto distinto de recursos do Azure** para cada um.
- **Infraestrutura como Código (IaC):**
 - Use **script** ou ferramentas de IaC para **especificar declarativamente** a configuração de cada ambiente.
 - Isso garante que o processo de criação seja **repetível** e permite criar novos ambientes sob demanda (ex: testes de segurança ou desempenho).
- **Suporte DevOps e CI/CD:** Azure DevOps (integração estreita com a nuvem), GitHub Actions (para repositórios GitHub).



Gerenciamento de Recursos: Ferramentas Disponíveis

- O Azure oferece diversas ferramentas, sendo comum o uso de uma **combinação** delas dependendo da tarefa.

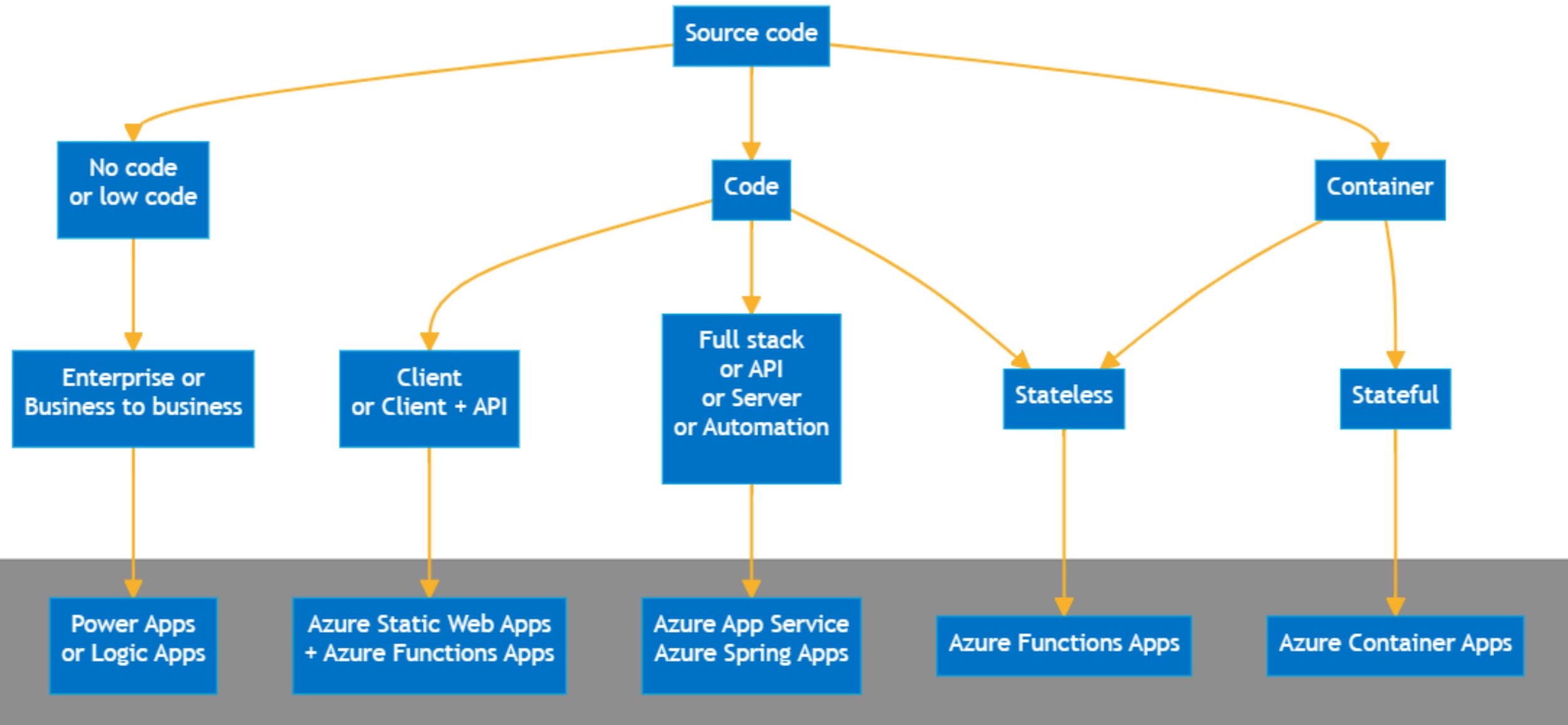
Tipo de Ferramenta	Uso Comum	Exemplo de Ferramenta
GUI (Interface Gráfica)	Criação de protótipos; orientação no processo de seleção de opções.	Portal do Azure, Extensão Ferramentas do Azure para VS Code
Linha de Comando (Script)	Automação de tarefas comuns (garante criação consistente); mais rápido que UI.	CLI do Azure, Azure PowerShell
IaC (Infraestrutura como Código)	Implantar e gerenciar recursos de forma declarativa . Garante implantação consistente e evita drift .	Bicep, Terraform, Ansible

Infraestrutura como Código (IaC)

- **Conceito IaC:** Processo de gerenciar e provisionar recursos por meio de **arquivos de configuração declarativos**.
 - Usa especificação de estado final para garantir que recursos sejam criados e configurados da mesma maneira todas as vezes.
 - Ferramentas IaC monitoram recursos para garantir que permaneçam no estado desejado.
- **Terraform:**
 - Ferramenta **open source** para provisionar e gerenciar infraestruturas de nuvem.
 - Codifica a infraestrutura em arquivos de configuração que descrevem a topologia.

Modelos de Hospedagem: Simplicidade vs Controle

- **Simplicidade vs Controle:** Consideração chave na escolha do serviço.
 - **Simplicidade (Totalmente Gerenciada):** Menos configuração, Azure gerencia runtime e infraestrutura (atualizações, patches).
 - Exemplos: Aplicativos Lógicos, Azure Functions (Sem servidor).
 - **Controle (Gerenciamento Total):** Controle total sobre a infraestrutura subjacente. Você é responsável por atualizações, patches e código.
 - Exemplos: Máquinas Virtuais do Azure, Serviço de Kubernetes do Azure.
- **Hospedagem Equilibrada:**
 - Balanceia a necessidade de simplicidade com a necessidade de controle.
 - Exemplos: Serviço de Aplicativo do Azure (hospedagem Web de serviço completo), Aplicativos de Contêiner do Azure (contêiner sem servidor).



Acordo de Nível de Serviço (SLA)

O Acordo de Nível de Serviço (SLA) é um compromisso formal entre o provedor de serviços de nuvem (como o Microsoft Azure) e o cliente, que define os níveis mínimos de qualidade, disponibilidade e desempenho que o serviço deve oferecer.



Measuring SLA's

The following table shows the potential cumulative downtime for various SLA levels.

SLA	Downtime per week	Downtime per month	Downtime per year
99%	1.68 hours	7.2 hours	3.65 days
99.9%	10.1 minutes	43.2 minutes	8.76 hours
99.95%	5 minutes	21.6 minutes	4.38 hours
99.99%	1.01 minutes	4.32 minutes	52.56 minutes
99.999%	6 seconds	25.9 seconds	5.26 minutes

O que o SLA garante?

- **Disponibilidade:** Percentual de tempo em que o serviço estará acessível e funcional. Por exemplo, um SLA de 99,9% significa que o serviço pode ficar indisponível por no máximo 43 minutos por mês.
- **Desempenho:** Tempo de resposta, capacidade de processamento e escalabilidade.
- **Suporte:** Tempo de resposta para chamados técnicos e canais de atendimento.
- **Compensações:** Se o SLA não for cumprido, o cliente pode receber créditos ou descontos conforme as políticas do provedor.

Por que o SLA é importante?

- Ajuda a **avaliar riscos** e escolher o serviço mais adequado para aplicações críticas.
- Garante **transparência** entre cliente e provedor.
- Serve como base para **auditorias e conformidade** com normas como ISO, LGPD e PCI-DSS.