Reproducible Developer Environments

MICHAEL PRICE

Microsoft

Welcome to CppCon 2022!

Join #visual_studio channel on CppCon Discord
https://aka.ms/cppcon/discord

- Meet the Microsoft C++ team
- Ask any questions
- Discuss the latest announcements

Take our survey
https://aka.ms/cppcon

Our sessions

Monday 12th
- **The Imperatives Must Go** – Victor Ciura
- **What's New in C++ 23** – Sy Brand
- **C++ Dependencies Don't Have to Be Painful** – Augustin Popa
- **How Microsoft Uses C++ to Deliver Office** – Zachary Henkel

Tuesday 13th
- **High-performance Load-time Implementation Selection** – Joe Bialek, Pranav Kant
- **C++ MythBusters** – Victor Ciura

Wednesday 14th
- **-memory-safe C++** - Jim Radigan

Thursday 15th
- **What's New for You in Visual Studio Code** – Marian Luparu, Sinem Akinci
- **Overcoming Embedded Development Tooling Challenges** – Marc Goodner
- **Reproducible Developer Environments** – Michael Price

Friday 16th
- **GitHub Features Every C++ Developer Should Know** – Michael Price
- **What's New in Visual Studio 2022** – Marian Luparu, Sy Brand
- **C++ Complexity (Keynote)** – Herb Sutter

What is a Developer Environment?

Some qualities

Affinity

Platform

Resources

Configuration

Interactivity

Difficulties Setting Up Development Environments

2022 ISO C++ Foundation Survey

Setting up a development environment from scratch (compiler, build system, IDE, …)

MAJOR PAIN POINT	MINOR PAIN POINT	NOT A SIGNIFICANT ISSUE
28%	43%	29%

But Why?

Because…

Procurement

Getting access to appropriate hardware and software can be time-consuming and expensive

Misconfiguration

Instructions can be ambiguous or misinterpreted, making configuration error-prone

Fragility

Small, seemingly uninteresting changes can break working development environments

Genericity

A generically useful development environment can be ill-suited for specific projects

Permanence

A permanent, single-instance development environment in the critical path of the developer loop is a bottleneck

Inconsistency

Accumulating minor variations in developer environments can harm collaborative efforts

Evaluating Possible Solutions

- Reproducible
- Purpose Built
- Managed
- Available
- Isolated
- Customizable
- Documented
- Versioned
- Procurable
- Responsive

My Personal Scorecard Rubric

1 of 3

	1	2	3
Reproducible	• Distinct provisioned environments are functional, but may have significant differences.	• Distinct provisioned environments are functionally equivalent, even for non-development productivity apps.	• Updates to existing provisioned environments can be applied to all provisioned environments, with the same results.
Purpose Built	• A provisioned environment is intended to be used for 5 or more distinct (but potentially similar) projects.	• A provisioned environment is intended to be used for 2-4 distinct (but potentially similar) projects.	• A provisioned environment is intended to be used for only 1 project. Distinct projects have distinct provisioned environments.
Managed	• Application of OS security patches and other system/tool updates are centrally managed. • Authentication & authorization for the environment are centrally managed.	• Corporate, government, and other regulations or policies are centrally managed and enforced. • Operational costs for all provisioned systems are monitored and traceable.	
Available	• A developer can use an environment without an active Internet connection.	• A developer can use an environment without an active network connection. • A developer can remotely use an environment, via a single application, such as a web browser or remote access application.	

My Personal Scorecard Rubric
2 of 3

	1	2	3
Isolated	• Usage of a development environment does not result in observable resource contention with other developer environments.	• Actions taken in a development environment do not alter the state or functionality of other developer environments.	
Customizable	• Developers can customize the environment to suit their personal tastes	• Developers can improve upon the shared development environment image/base without involving outside departments.	
Documented	• Steps to recreate a developer environment from scratch are documented. • … and kept up-to-date	• Documentation is “executable” (e.g. a script or input files to tool).	

My Personal Scorecard Rubric

3 of 3

	1	2	3
Versioned	• The developer environment is versioned, with a clear indication what environment versions work with what repository versions.	• The developer environment is versioned alongside the repository that it is intended for.	
Procurable	• A developer environment can be acquired and configured in less than 1 week (on average).	• A developer environment can be acquired and configured in less than 24 hours (on average)	• A developer environment can be acquired and configured in less than 30 minutes.
Responsive	• The developer environment consistently responds to user input without noticeable delay.		

Example Scorecard – Traditional Corporate Developer Environments

Contoso			
	1 Reproducible	1 Purpose Built	4 Managed
2 Available	2 Isolated	1 Customizable	1 Documented
0 Versioned	1 Procurable	1 Responsive	14 Overall

Microsoft Dev Box

Transforming the app development ecosystem

Enable self-service development with Microsoft Dev Box

Provide developers with self-service access to high-performance, cloud-based workstations preconfigured and ready-to-code for specific projects

Transform the developer workstation

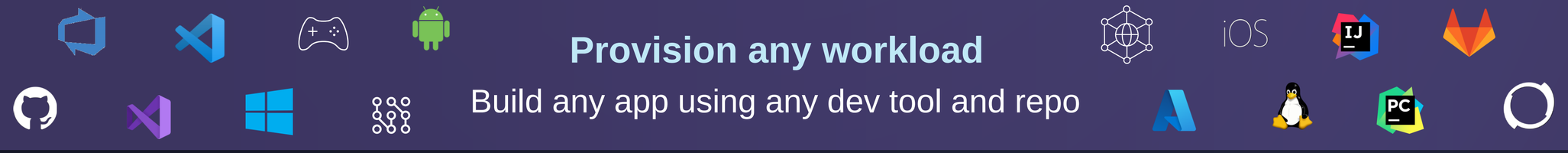
Replace curated, long-running developer desktops with ephemeral resources that spin up on-demand for any dev workload

Streamline dev team collaboration

Customize configurations to support every developer on the team with the tools and resources they need to succeed—wherever they are

Stay secure and compliant

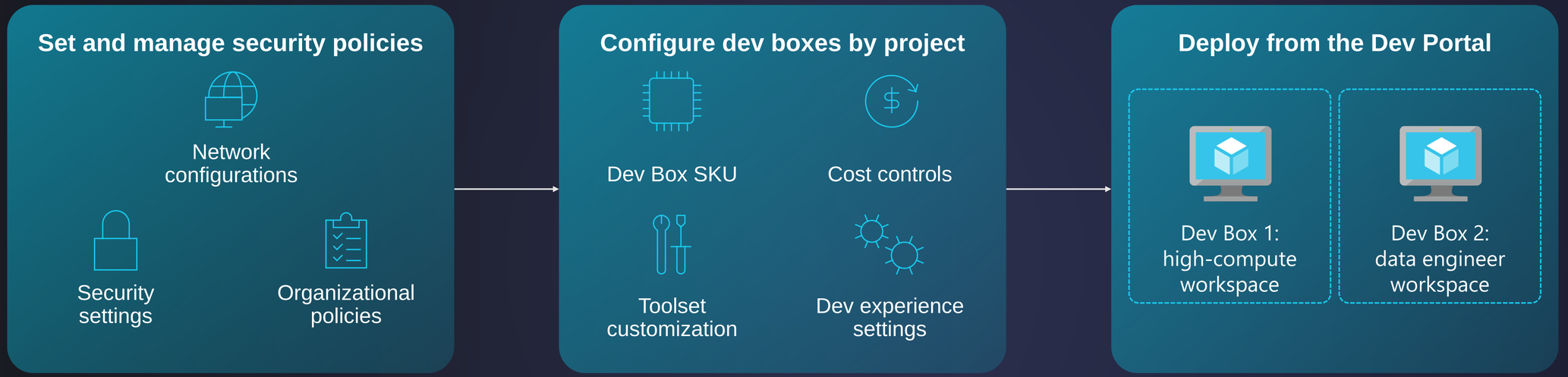
Empower developers with flexible, self-service workstations while keeping costs and risks low by centralizing control under IT teams



Provision any workload

Build any app using any dev tool and repo

How different roles use Microsoft Dev Box



IT / Dev Infra Teams
Manage Dev Boxes via Intune and Microsoft Endpoint Manager

Dev Teams
Create pools of Dev Boxes tailored to developers' projects and tasks

Developers
Deploy the Dev Boxes they need to work on their current tasks and projects

Control costs with consumption-based pricing

Pay only for the compute and storage you use

Storage meter measures hours of provisioned disk from creation to deletion

Compute meter measures hours of active vCore usage from Dev Box start to stop

Start free during public preview

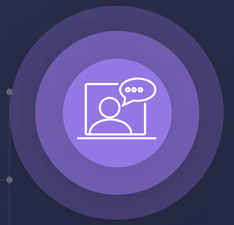
Every month during public preview, the first 15 hours of the dev box 8 vCPU and 32 GiB Memory SKU are free, along with the first 365 hours of the dev box Storage SSD 512 GiB SKU

READY TO GET STARTED?

Microsoft Dev Box is in Public Preview

Read the Microsoft Dev Box launch blog to learn more

Request a demo to see Microsoft Dev Box in action



Start using Microsoft Dev Box today

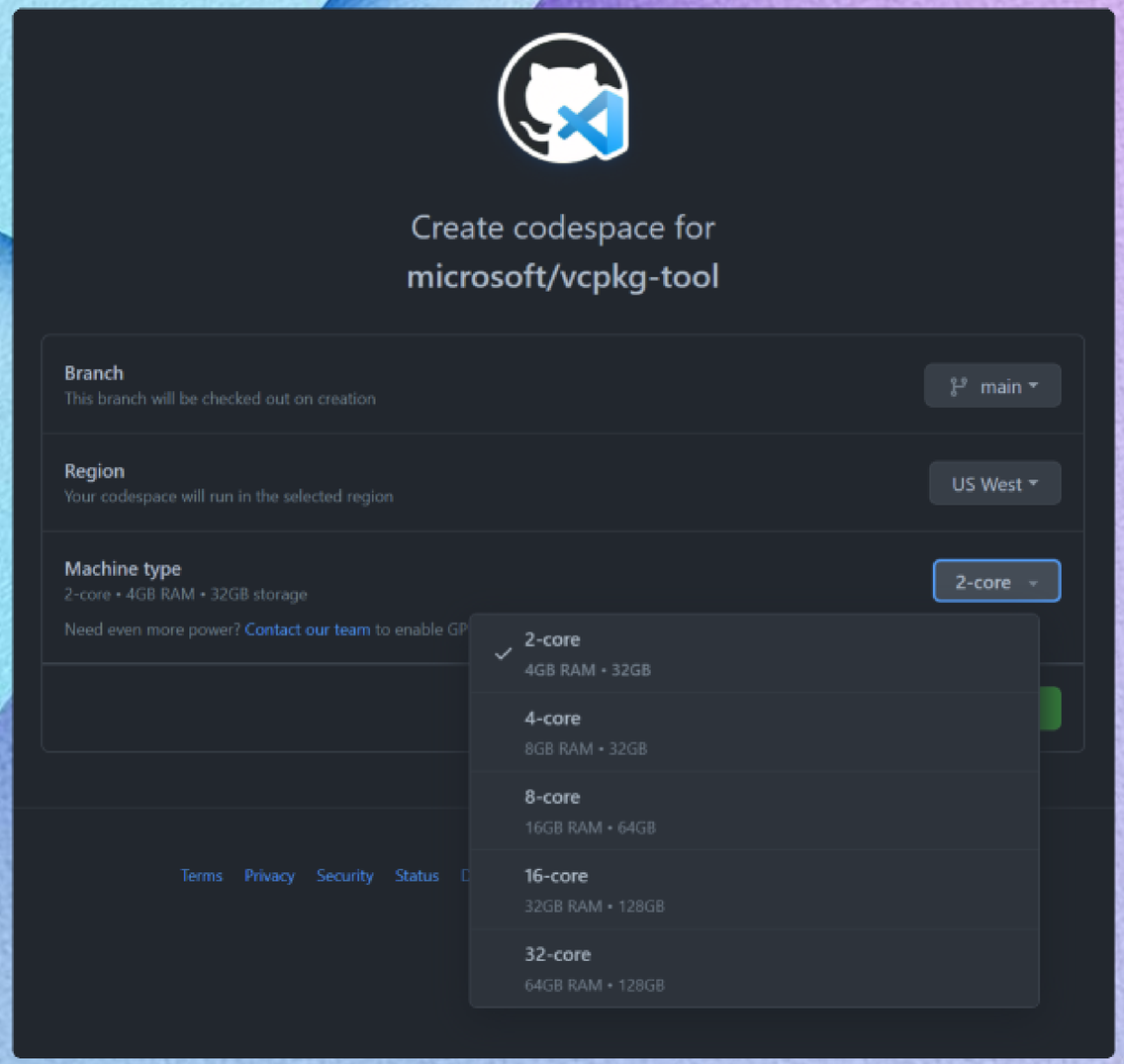
Example Scorecard – Traditional Corporate Developer Environments

Microsoft Dev Box	3 Reproducible	2 Purpose Built	4 Managed
2 Available	3 Isolated	3* Customizable	0 Documented
1 Versioned	2 Procurable	1 Responsive	21 Overall

Microsoft Dev Box Demonstration

GitHub Codespaces

- On-demand, container-based, cloud development environments
- Persistent state across work sessions
- Limited portability between different VM SKUs
- Customizable environments with dev containers
- Can be prebuilt from GitHub Actions



Example Scorecard – Traditional Corporate Developer Environments

GitHub Codespaces			
3	3	3	3
Reproducible	Purpose Built	Managed	
2	3	3	4
Available	Isolated	Customizable	Documented
2	3	1	27
Versioned	Procurable	Responsive	Overall

GitHub Codespaces Demonstration

Dev Box vs Codespaces… what should you use?

Microsoft Dev Box

- Windows development
- Restrictions from corporate policies or regulation
- Need access to OS desktop
- Need to have “warm” environments

GitHub Codespaces

- Linux development (currently)
- Contributors might work on many (dozens) of projects
- Code hosted on GitHub
- No IT or DevOps team to manage cloud infrastructure

Make Your Own Scorecard

Allocate points for what's important for you and compare different solutions!!!

Happy Coding!

Microsoft

Enjoy the rest of the conference!

Join #visual_studio channel on CppCon Discord
https://aka.ms/cppcon/discord

- Meet the Microsoft C++ team
- Ask any questions
- Discuss the latest announcements

Take our survey
https://aka.ms/cppcon

Questions & Answers