Kubernetes Resource Scaling via Batch Node Conversion on the Anvil Supercomputer

Erik Gough, LJ Lumas

Rosen Center for Advanced Computing (RCAC)

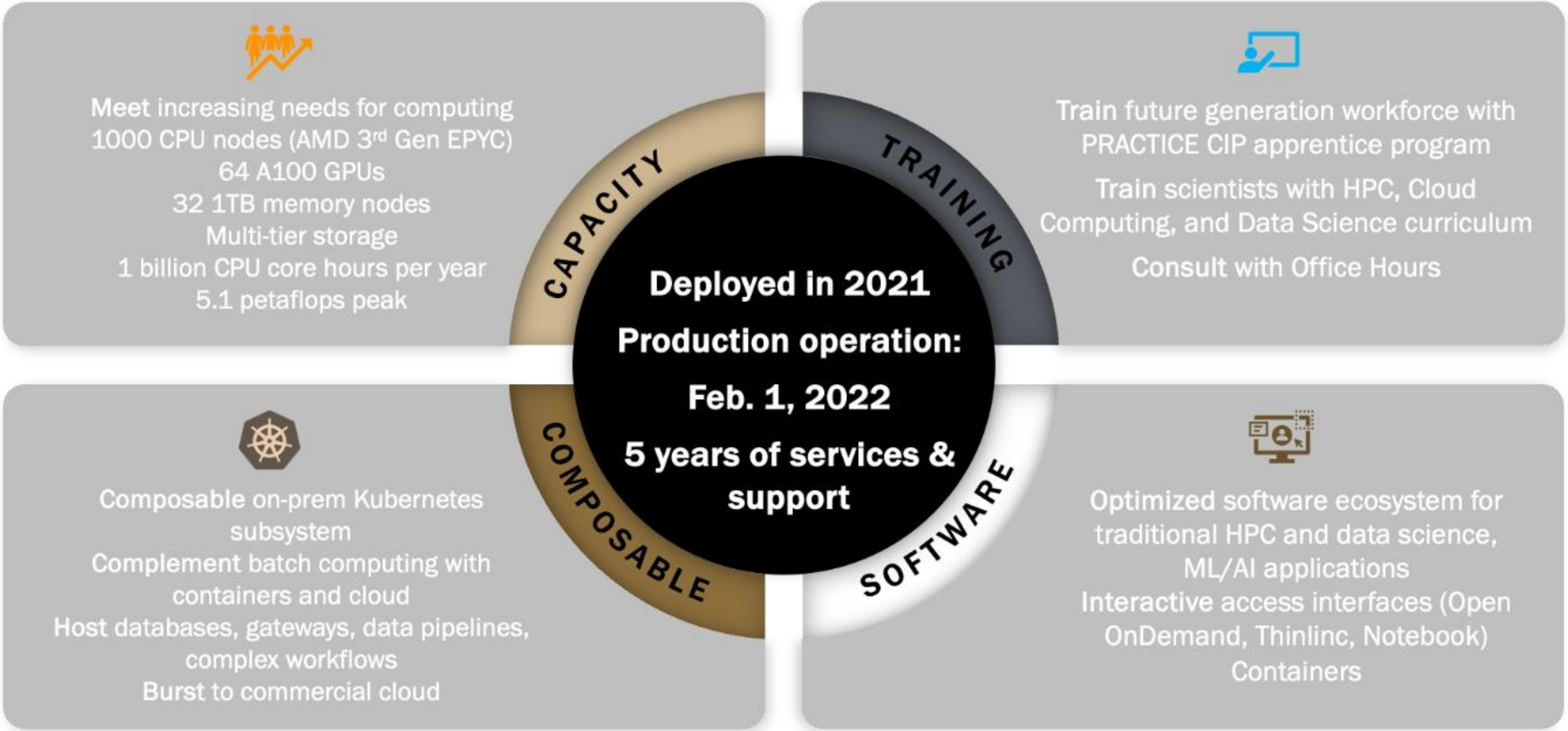
HPCSYSPROS24

11/22/24

PURDUE UNIVERSITY®

Anvil Introduction

Anvil - A National Composable Advanced Computational Resource for the Future of Science and Engineering



Deployed in 2021

Production operation:

Feb. 1, 2022

5 years of services & support

CAPACITY

Meet increasing needs for computing
1000 CPU nodes (AMD 3rd Gen EPYC)
64 A100 GPUs
32 1TB memory nodes
Multi-tier storage
1 billion CPU core hours per year
5.1 petaflops peak

TRAINING

Train future generation workforce with PRACTICE CIP apprentice program

Train scientists with HPC, Cloud Computing, and Data Science curriculum

Consult with Office Hours

COMPOSABLE

Composable on-prem Kubernetes subsystem
Complement batch computing with containers and cloud
Host databases, gateways, data pipelines, complex workflows
Burst to commercial cloud

SOFTWARE

Optimized software ecosystem for traditional HPC and data science, ML/AI applications
Interactive access interfaces (Open OnDemand, Thinlinc, Notebook)
Containers

The Problem

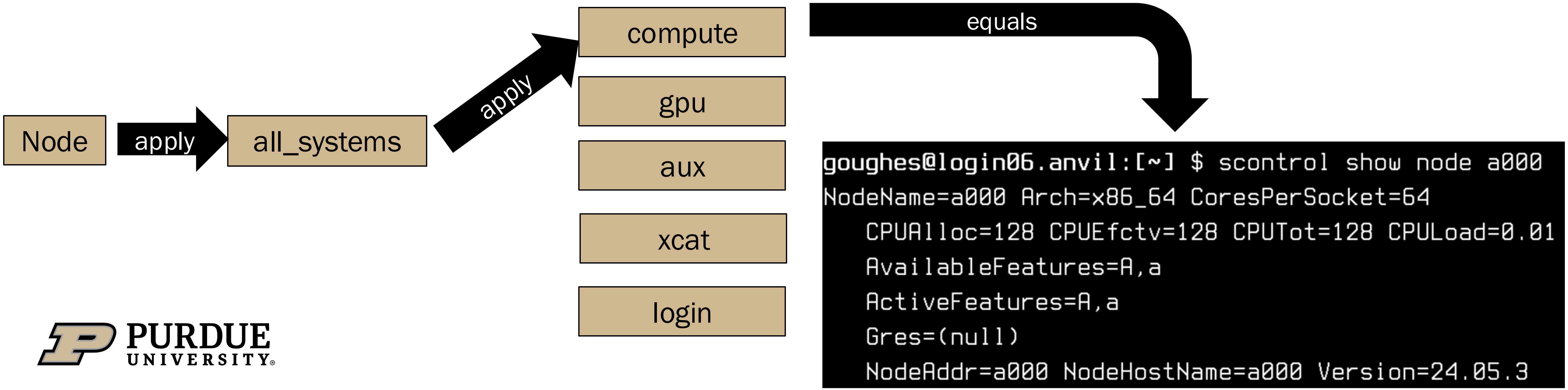
Kubernetes, so hot right now

- The Anvil Kubernetes cluster currently consists of 512 CPU cores and 4 A100 GPUs
- We see an ever-increasing demand for Kubernetes infrastructure
  - 12 Science Gateways with various scaling needs (CPU + GPU)
  - Anvil Notebook Service – JupyterHub platform (CPU + GPU)
  - Anvil GPT – On-prem GenAI/LLM platform (GPU)
- Limited resources available on Kubernetes clusters compared to Anvil HPC system
- Demand spikes from workshops and training sessions often exceed Kubernetes resource capacity

Configuration Management

xCAT + stateless images + masterless Puppet (Jason St. John, HPCSYSPROS18)

- On Anvil, we use xCAT, Puppet and Git repos as the source of truth for node configuration
- xCAT images are built and run in RAM on each node (stateless)
- Each node acts as its own Puppet master
  - /etc/puppet is symlinked to a local copy of the cluster's Git repo
  - A role is defined in /etc/puppet-role
  - A run_puppet script applies the puppet configuration from two modules (all_systems + <role>)



```
goughes@login06.anvil:[~] $ scontrol show node a000
NodeName=a000 Arch=x86_64 CoresPerSocket=64
   CPUAlloc=128 CPUEfctv=128 CPUTot=128 CPULoad=0.01
   AvailableFeatures=A,a
   ActiveFeatures=A,a
   Gres=(null)
   NodeAddr=a000 NodeHostName=a000 Version=24.05.3
```

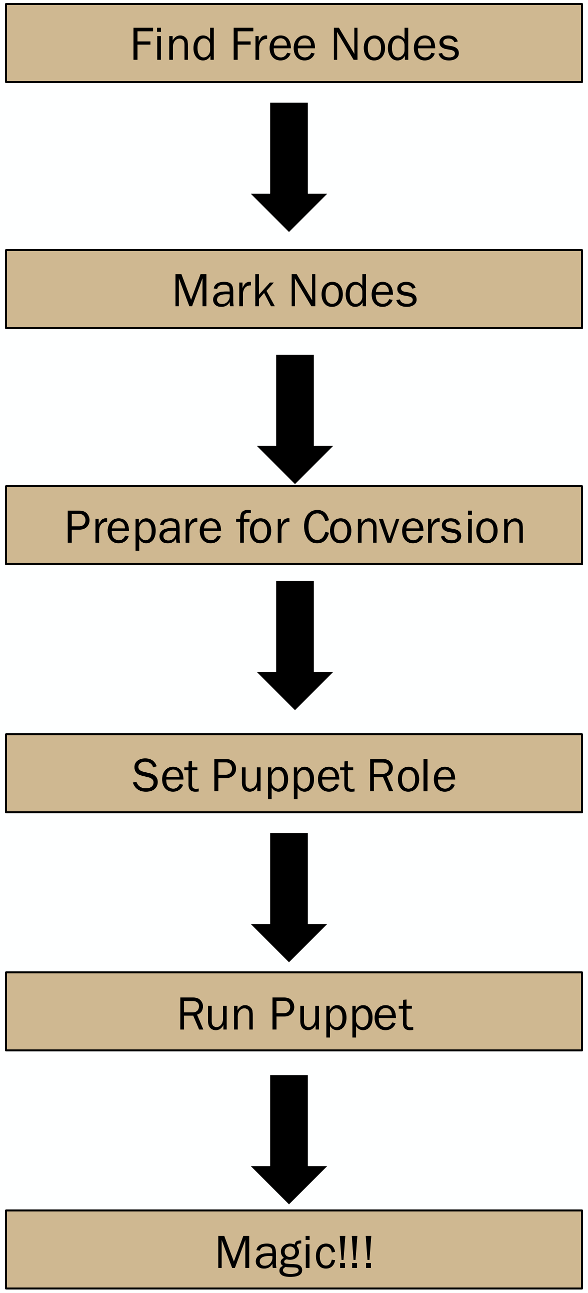
PURDUE UNIVERSITY®

How it works – Batch to Kubernetes

We created a separate "cloud_compute" puppet role to perform required steps for transitioning a "compute" or "gpu" node to k8s.

- **Find Free nodes:** Query Slurm to find nodes in IDLE state.
- **Mark Nodes**: Mark nodes DOWN in Slurm, preventing them from being used for other batch workloads.
- **Prepare for Conversion**: Perform preliminary adjustments to node settings, including handling specialized hardware like GPUs, to ensure compatibility with Kubernetes.
- **Set Puppet Role**: Assign the appropriate "cloud_compute" role to the node.
- **Run Puppet:** Initiate a puppet run using the new node role.
- **Magic*:** Puppet reconfigures the node to conform to requirements for cloud compute nodes.

* refer to next slide



The Magic

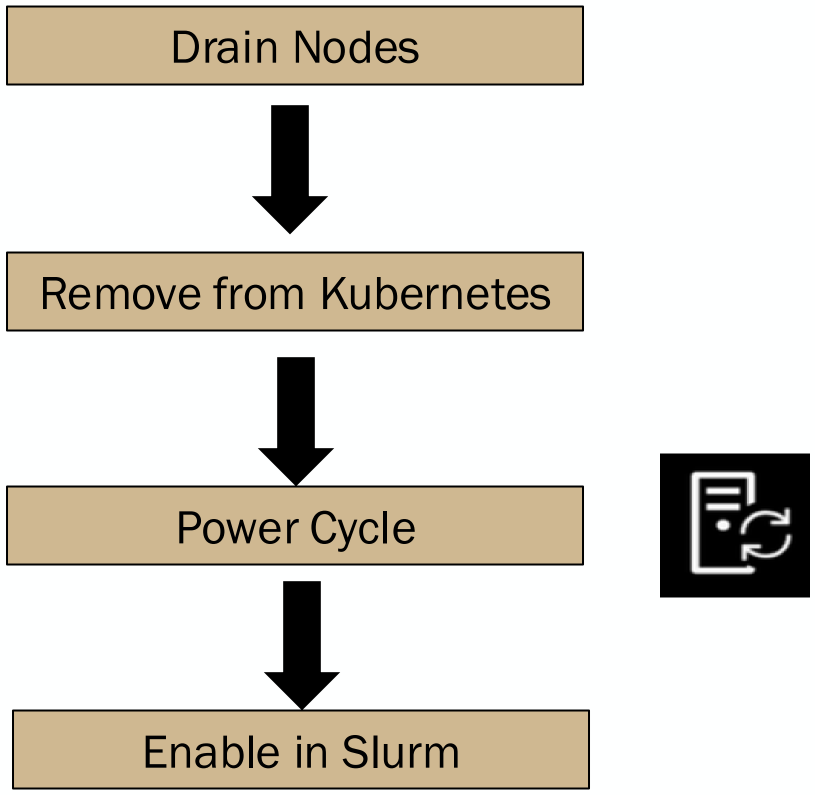
What actually happens in Puppet

- Install our Container Runtime Interface (Docker)
- Modify firewall rules (enable forward chain, block access to NFS, etc.)
- Stop unnecessary or conflicting services
  - slurmd
  - munged
  - gpfs (as needed)
  - PCP
  - Zabbix
  - node_exporter
- Start Docker
- Run Kubernetes registration command with appropriate taints or labels
- Kubernetes components are deployed (CNI, ingress, other DaemonSets)
- Nodes are ready for scheduling in Kubernetes

How it works – Kubernetes to Batch

K.I.S.S.

- **Drain Kubernetes Nodes:** Safely stop and remove workloads from the nodes to prepare them for conversion to batch processing.
- **Delete from Kubernetes:** Remove nodes from the Kubernetes cluster.
- **Power Cycle Nodes:** Restart the nodes to reset their configuration and clear any remaining Kubernetes dependencies.
- **Reclaim Nodes for Batch:** Set nodes back to IDLE in Slurm, making them available for scheduling.



Outcomes

Batch node conversion has enabled scaling for several research groups

- NanoHUB STARS Workshop
  - Supported 75 participants using containers with 4C and 16GB RAM
- CMS FastML Workshop (Anvil Notebook Service)
  - Supported 25 participants using containers with 4C and 32GB RAM
- CyberFACES (NSF CyberTraining)
  - Custom JupyterHUB supporting 100s of participants
- Purdue DataMine (Anvil Notebook Service, 2025)
  - 1200+ students currently using Anvil batch to launch notebooks

Future Work

What's next?

- Nodes are currently converted manually based on need
- Dynamic node allocation based on queue pressure
  - Query Kubernetes API
  - Automatically provision nodes with work presented in this talk
- Dynamic node deallocation
  - Automatic cordoning or draining of nodes
  - Reallocation into Slurm

Thanks!
Questions?

This material is based upon work supported by the National Science Foundation under Grant No. 2232872. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

PURDUE
UNIVERSITY®