GUROBI OPTIMIZATION

GUROBI 9.5

Launch Event

December 2 & 3, 2021

Speakers

Dr. Tobias Achterberg

Dr. Michel Jaczynski

What’s New in Gurobi 9.5

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

GUROBI
OPTIMIZATION

Gurobi 9.5

Performance Summary

Performance improvements compared to Gurobi 9.1

Algorithm	Overall speed-up	On >100sec models
Concurrent LP	14%	54%
Primal simplex	23%	43%
Dual simplex	20%	43%
Barrier	18%	56%
MILP	15%	27%
Convex MIQP	30%	68%
Convex MIQCP	33%	78%
Non-convex MIQCP	3.0x	7.5x

GUROBI
OPTIMIZATION

Gurobi 9.5

Performance Summary

Performance improvements compared to Gurobi 9.1

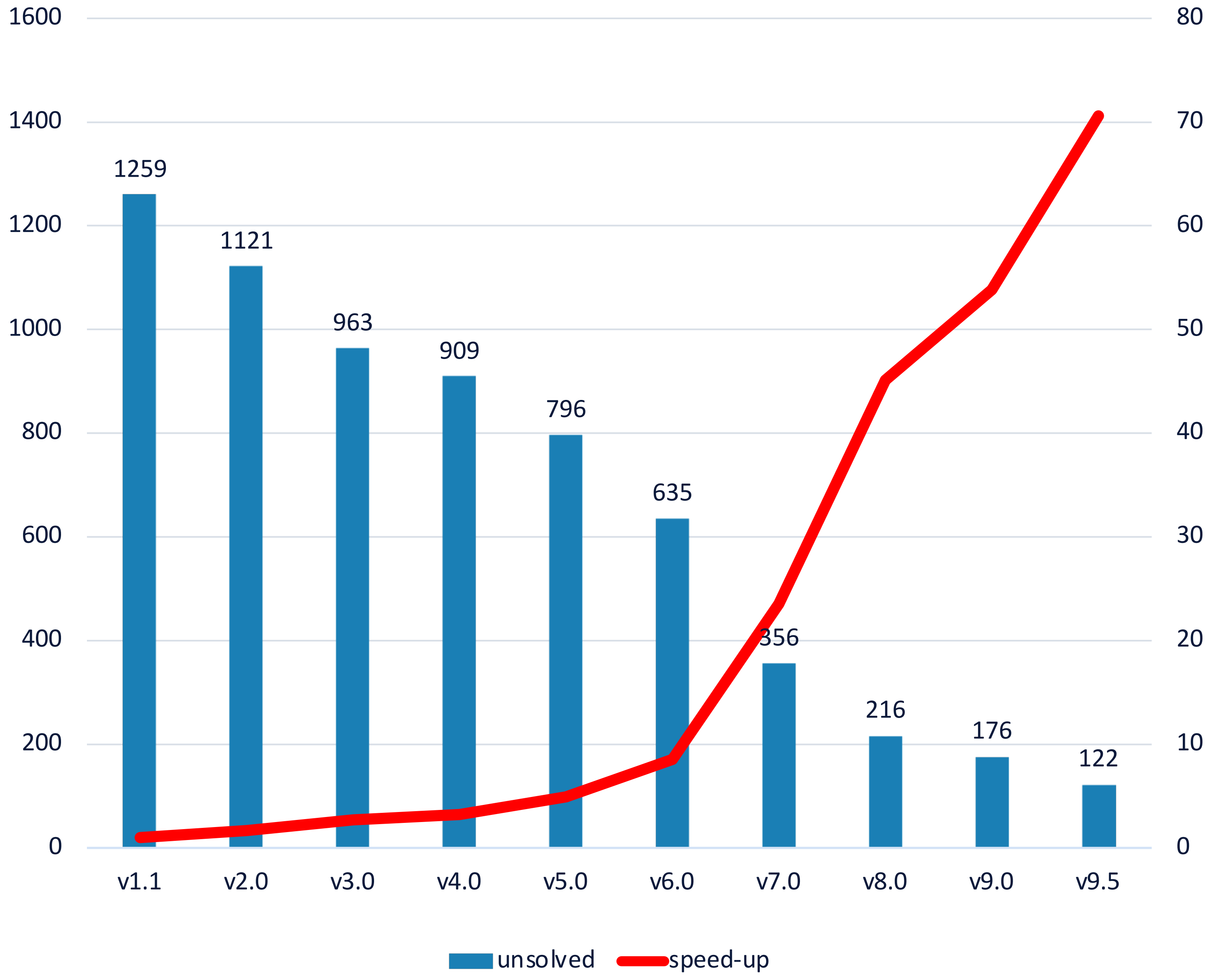
Algorithm	Overall speed-up	On >100sec models
Concurrent LP	14%	54%
Primal simplex	23%	43%
Dual simplex	20%	43%
Barrier	18%	56%
MILP	15%	27%
Convex MIQP	30%	68%
Convex MIQCP	33%	78%
Non-convex MIQCP	3.0x	7.5x

GUROBI
OPTIMIZATION

MILP

Performance Evolution

Comparison of Gurobi Versions (PAR-10)



SOS Constraint Encodings

SOS constraints

- $SOS1(x_1, \ldots, x_n)$: at most one variable of $x_1, \ldots, x_n$ can be non-zero
- $SOS2(x_1, \ldots, x_n)$: at most two variables can be non-zero, and they must be adjacent

SOS constraints can be handled by branching

If all variables have finite bounds: can use MIP model

- Example: "multiple-choice" model for SOS1:

$$\begin{aligned}&\sum z_j \leq 1\\ &l_j z_j \leq x_j \leq u_j z_j \text{ for all } j\\ &z_j \text{ binary for all } j\end{aligned}$$

Several encodings in the literature (see Huchette, Vielma 2018):

- Linear size: multiple-choice, incremental
- Logarithmic size: various representations based on Binary Gray codes

Those encodings usually lead to the same LP relaxation value

- But differ in their size and strength within a branch-and-bound algorithm

SOS Constraint Encodings

Prior versions of Gurobi

- Multiple-choice formulation may be applied
- Depends on the magnitude of the bounds of the variables in the SOS
- Controlled via parameters “PreSOS1BigM” and “PreSOS2BigM”

New parameters “PreSOS1Encoding” and “PreSOS2Encoding” in Gurobi 9.5

- -1: Default (automatic)
- 0: Multiple-choice (linear size)
- 1: Incremental (linear size)
- 2: Binary Gray code (logarithmic size)
- 3: ZigZag integer code (logarithmic size)

Performance impact:

- 6% faster on all SOS models in our MIP test set
- 14% faster on SOS models that take more than 10 seconds

GUROBI
OPTIMIZATION

Gurobi 9.5

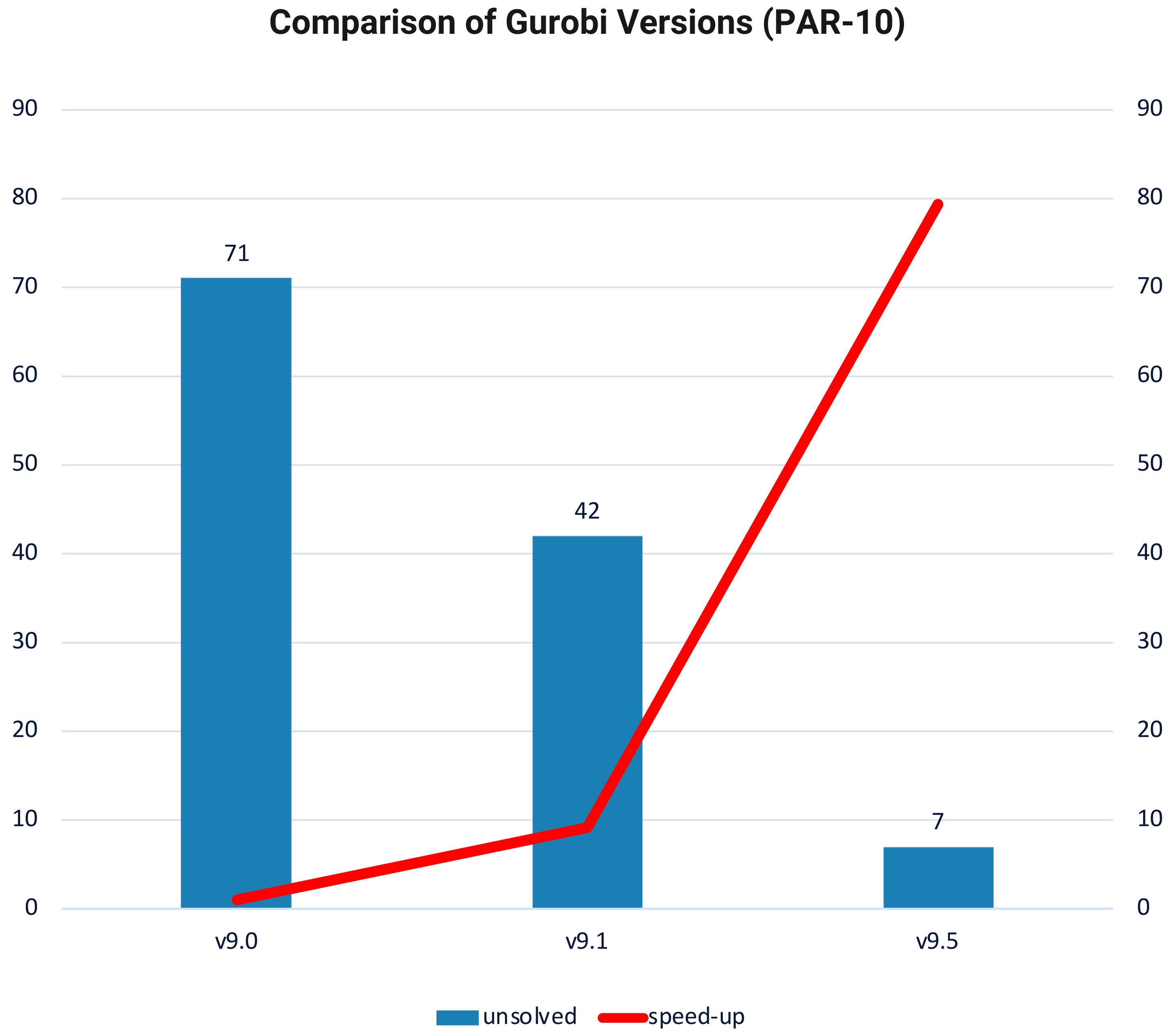
Performance Summary

Performance improvements compared to Gurobi 9.1

Algorithm	Overall speed-up	On >100sec models
Concurrent LP	14%	54%
Primal simplex	23%	43%
Dual simplex	20%	43%
Barrier	18%	56%
MILP	15%	27%
Convex MIQP	30%	68%
Convex MIQCP	33%	78%
Non-convex MIQCP	3.0x	7.5x

Non-convex MIQCP

Performance Evolution



NLP Heuristic

Often hard to find feasible solutions for non-convex MIQCPs

- In particular, for non-convex quadratic equations with continuous variables
- Only strategy in Gurobi 9.1: branch until domains are tiny

New in Gurobi 9.5: NLP heuristic

- Use fix-and-propagate to fix integer variables
- Call non-linear interior point algorithm on remaining continuous model

Non-linear interior point algorithm

- Uses a parallel, direct, sparse LBL^T linear system solver
- Quadratic local convergence
- Heuristics to help achieve global convergence
- Essentially: IPOPT restricted to its skeleton

GUROBI
OPTIMIZATION

NLP Heuristic

Performance Impact

Performance improvements to Gurobi 9.5 without NLP heuristic

Time to ...	Overall speed-up	On >100sec models
proven optimality	19%	41%
1% gap	21%	38%
10% gap	45%	97%
first feasible solution	90%	144%

Number of models where we don’t find any feasible solution is reduced from 79 to 60

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

GUROBI
OPTIMIZATION

LP Warm Starts with Presolve

Two options to set warm start information for LP

- Specify solution via “PStart” and “DStart” attributes; or
- Specify basis via “VBasis” and “CBasis” attributes
  - Note: modifying an LP after solving it retains the basis

Parameter “LPWarmStart”

- 0: Ignore warm start information
- 1: Disable presolve if warm start exists (default)
  - Regardless of presolve setting
  - Prefer basis over start vectors
- 2: Apply presolve if enabled by presolve setting
  - Prefer start vectors over basis
  - Approach:
    1. If only basis is given: apply simplex with zero iteration limit to compute start vectors
    2. Apply presolve
    3. Crush start vectors to use them for warm start on presolved model

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

General Constraints for Norms

Support for norm constraints

```
import gurobipy as gp
...
x = model.addVars(10)
y = model.addVar()
model.addConstr(y == gp.norm(x, n))
```

- Specifies that y should be equal to the n-norm of vector x
  - $n = 0$: $y = \|x\|_0 = \left|\{j \,|\, x_j \neq 0\}\right|$ (0 pseudo norm)
  - $n = 1$: $y = \|x\|_1 = \sum |x_j|$ (sum norm)
  - $n = 2$: $y = \|x\|_2 = \sqrt{\sum x_j^2}$ (Euclidean norm)
  - $n = \infty$: $y = \|x\|_\infty = \max |x_j|$ (maximum norm)

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Deterministic Work Measure

Wall clock run-time of an optimization is non-deterministic

- Two runs with same data, same parameters, same machine will have slightly different run-time
- Consequence: “TimeLimit” parameter leads to non-deterministic results
  - Final solution may be different in a second run

New “Work” attribute provides deterministic work measure

- Two runs with same data, same parameters, same machine will always result in same amount of “Work”
- Consequence: “WorkLimit” parameter leads to deterministic results

Deterministic Work Measure

Work is displayed in log file

```
Explored 3254 nodes (102518 simplex iterations) in 1.18 seconds (0.96 work units)
Thread count was 8 (of 8 available processors)

Solution count 10: 1158 1159 1195 ... 2548

Optimal solution found (tolerance 1.00e-04)
Best objective 1.158000000000e+03, best bound 1.158000000000e+03, gap 0.0000%
```

Very roughly: 1 deterministic work unit ≅ 1 second

- Heavily depends on machine, model, and algorithm
  - Can vary pretty heavily (factors of 3-5 are not uncommon)
- But work/time ratio pretty stable on models of same problem class and similar size
- You need to experiment with your models on your hardware!

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Memory Limit Parameter

Limit the total amount of memory Gurobi is allowed to use

```
import gurobipy as gp
p = {'MemLimit’: 1.5}    # max 1.5 GB
with gp.Env(params=p) as env:
  model = gp.read(‘model.mps’, env=env)
  model.optimize()
```

Can only be set in the initial environment

- All sub-environments inherit the limit
- Cannot be modified anymore after initial environment was started

Note: memory limit hits can be non-deterministic in parallel algorithms

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

GUROBI
OPTIMIZATION

Solution Pool

Solutions found during a MIP search are stored in a pool

- Contains set of different solutions
  - Each pair of solutions differs on at least one integer variable
- “PoolSolutions”: maximum number of solutions to store
- “PoolGap”, “PoolGapAbs”: discard solutions beyond this gap
- “PoolSearchMode”: search for more or all feasible solutions

New in Gurobi 9.5: the “PoolIgnore” attribute

- If set to 1 for a certain variable x_j, ignores x_j in identity checks
  - Each pair of solutions differs on at least one integer variable *for which the “PoolIgnore” attribute is not set*
- Use case:
  - Ignore auxiliary variables
  - Only focus on differences in main decision variables

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Guiding the IIS Computation

computeIIS() finds an irreducible infeasible subsystem (IIS) for an infeasible model

- Subsystem is infeasible
- Any further removal of a constraint or a bound makes the IIS feasible

New attributes

- IISLbForce, IISUbForce
  - = - 1: default (let IIS algorithm decide whether to include bound)
  - = 0: force the bound to be not in the IIS
  - = 1: force the bound to be in the IIS
- IISConstrForce, IISSOSForce, IISQConstrForce, IISGenConstrForce
  - Same for constraints

Notes

- If constraints or bounds forced out of the IIS, remaining model might be feasible
- If constraints or bounds forced into IIS, may lead to “IIS” that is not irreducible

Guiding the IIS Computation

Typical use case for IIS force flags

- You have a base model that you know is feasible
- A derived model is created by
  - adding constraints and variables
  - tightening right hand sides and bounds of existing constraints and variables
- Resulting model is infeasible
- You are only interested which of the additional changes caused the infeasibility

Solution

- Set IIS force flags of base model to 1
- computeIIS() typically runs much faster
  - Does not need to check whether base model constraints and bounds can be removed from the infeasible subsystem

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Callback Enhancements

Solutions can now be passed to Gurobi from the MIP and MIPSOL callbacks as well

- E.g., if you can improve a solution provided by Gurobi as MIPSOL, you can directly add it back with cbUseSolution()
- No need to store it internally and wait for the next MIPNODE callback

Callback invoked from the NoRel heuristic

- Allows to add lazy constraints to cut off solutions
- Can signal to exit NoRel and proceed to the next phase of optimization

Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks

- Current phases:
  - NOREL: inside NoRel heuristic
  - SEARCH: regular tree search
  - IMPROVE: solution improvement phase

Callback Enhancements

Callback query code OPENSCENARIOS

- In a multi-scenario solve, returns number of scenarios that are not yet solved

Support for callbacks during an IIS computation

- Callback query codes "ConstrMin", "ConstrMax", "BoundMin", "BoundMax“
    - Obtain the current lower and upper bound on the final IIS size
- Callback query codes "ConstrGuess" and "BoundGuess“
    - Obtain the current estimate on the final IIS size

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Improvements in gurobipy

Support for Python 3.10

Optional dependencies in PyPI.org

- Core functionality of gurobipy
  - No external dependencies
  - `pip install gurobipy`
- Extended functionality with matrix-friendly API
  - Depends on NumPy and SciPy
  - `pip install gurobipy[matrixapi]`
  - Automatically installs NumPy and SciPy

Improved indexing of MVar and MConstr objects

- Selecting elements of such objects now returns a scalar object
- These ndarray-like classes behave more closely to NumPy ndarrays
  - `x = m.addVars(10)` → `x[3]` is now a Var object
    - previously, was an MVar of shape (1,)

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

GUROBI
OPTIMIZATION

Tuning Tool Enhancements

Tuning tool is used to automatically find good solver parameter settings for a given set of models

- Tune parameters on a set of instances of the same problem class
- Use parameters on new instances of the same class to solve them faster

New parameters for tuning tool

- “TuneMetric”: tune for best average or best worst-case performance
- “TuneTargetTime”: stop tuning if a parameter combination has been found that reaches a given target solve time
- “TuneTargetMIPGap”: stop tuning if a parameter combination has been found that reaches a given target MIPGap within the allowed time limit

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Universal2 Port for Mac

Macs available with two different CPU types

- x64 – Intel or AMD
- Apple silicon – e.g., Apple M1

Different machine instructions needed for each type of CPU

Universal2 is Apple’s approach to include both instruction sets in single binary

- OS automatically selects the right code path, depending on CPU

Gurobi’s Mac universal2 port runs on both x64 and Apple silicon

- Old mac64 port no longer provided
- Runs on macOS 10.14 (Mojave) and later versions

Gurobi 9.5

Summary

Performance improvements

- Significant performance improvements on all problem types
- Lift-and-project cuts
- SOS constraint encodings
- NLP heuristic for non-convex quadratic models

Support for LP warm starts in combination with presolve

New file formats to write the dual formulation of an LP problem

New “norm" general constraints

Deterministic work measure

Memory limit parameter

User control of solution pool

User control to guide IIS computation

Callback enhancements

- Pass user solutions to Gurobi from the MIP and MIPSOL callbacks
- Callback invoked from the NoRel heuristic
- Callback query code PHASE for MIP, MIPNODE, and MIPSOL callbacks
- Callback query code OPENSCENARIOS
- Support for callbacks during an IIS computation

Additional attributes

- OpenNodeCount
- ConcurrentWinMethod
- MaxVio

Improvements in gurobipy

Tuning tool enhancements

Universal2 port for Mac

Compute Server and Cluster Manager enhancements

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Web License Service for container deployment

Platform Enhancements in Gurobi 9.5

Gurobi R&D Team

December 2021

GUROBI OPTIMIZATION

Platform Enhancements in Gurobi 9.5

Compute Server/Cluster Manager

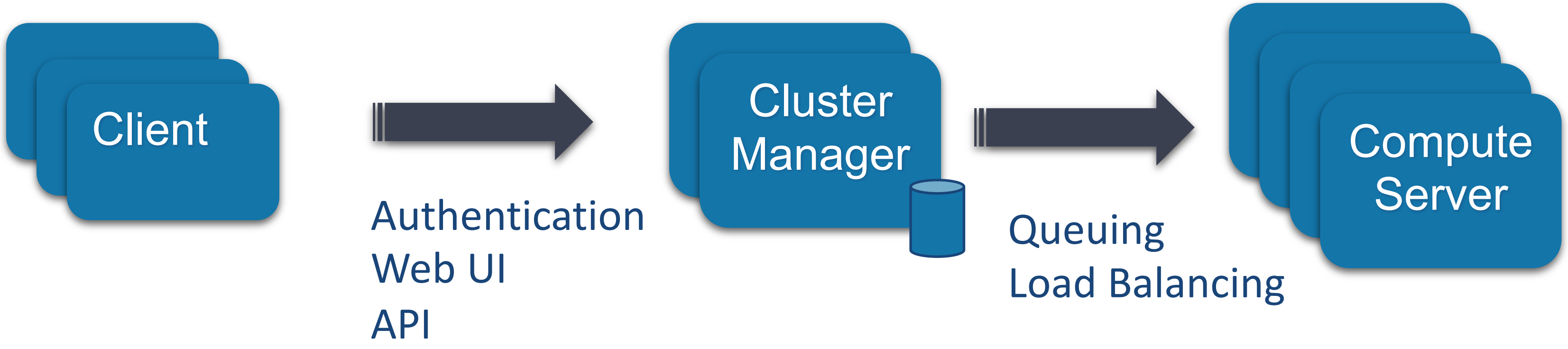
- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Gurobi on Containerized Environments

- Docker Hub Predefined Images
- Web License Service
- Best Practices for Kubernetes

Demo

Compute Server and Cluster Manager

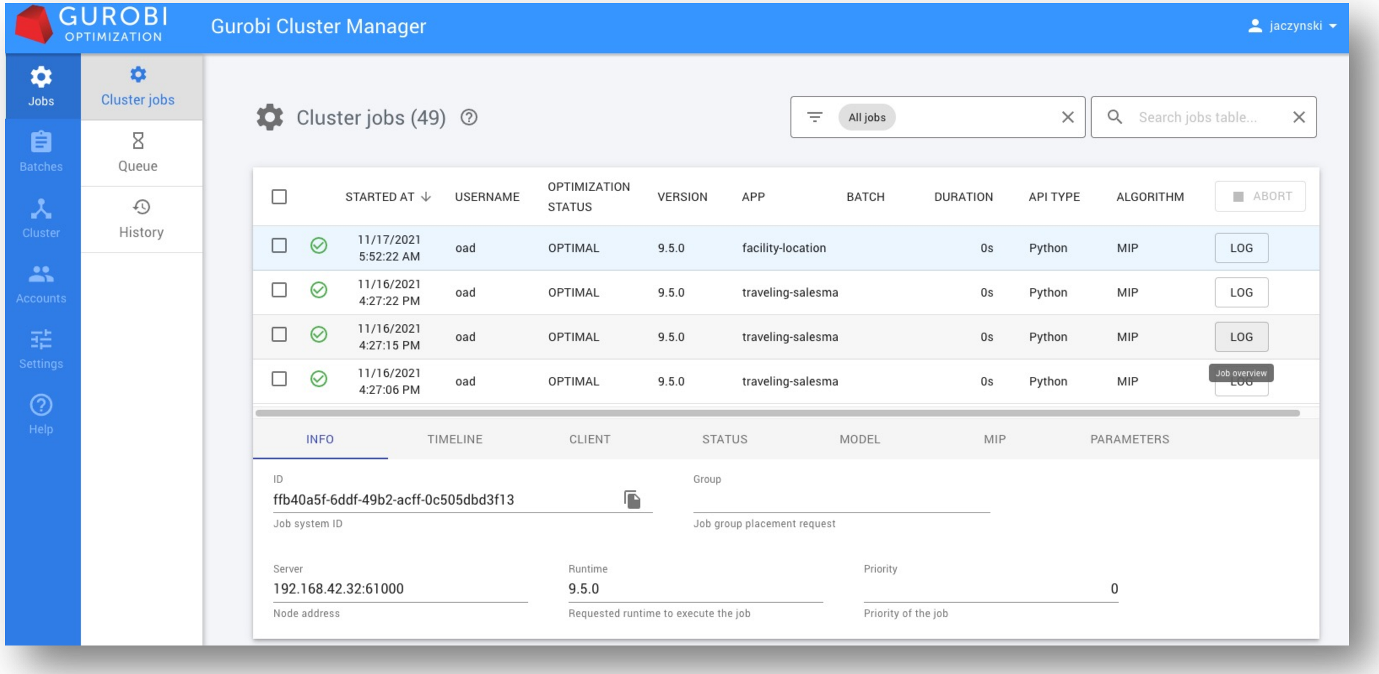


- Seamlessly offload optimization tasks to one or more servers
- Common use cases:
  - Share resources across users, teams and applications
  - Access more powerful machines
  - Ensure high availability with multiple nodes
  - Build modern applications, service-oriented architecture

Cluster Manager

- For IT
    - Modern and scalable architecture
    - Security
    - Web UI to manage the cluster

- For OR Developers
    - Web UI to monitor jobs
    - Job history
    - Batch optimization



□		STARTED AT	USERNAME	OPTIMIZATION STATUS	VERSION	APP	BATCH	DURATION	API TYPE	ALGORITHM	ABORT
□	✓	11/17/2021 5:52:22 AM	oad	OPTIMAL	9.5.0	facility-location		0s	Python	MIP	LOG
□	✓	11/16/2021 4:27:22 PM	oad	OPTIMAL	9.5.0	traveling-salesma		0s	Python	MIP	LOG
□	✓	11/16/2021 4:27:15 PM	oad	OPTIMAL	9.5.0	traveling-salesma		0s	Python	MIP	LOG
□	✓	11/16/2021 4:27:06 PM	oad	OPTIMAL	9.5.0	traveling-salesma		0s	Python	MIP	LOG

ID
ffb40a5f-6ddf-49b2-acff-0c505dbd3f13
Job system ID

Group

Job group placement request

Server
192.168.42.32:61000
Node address

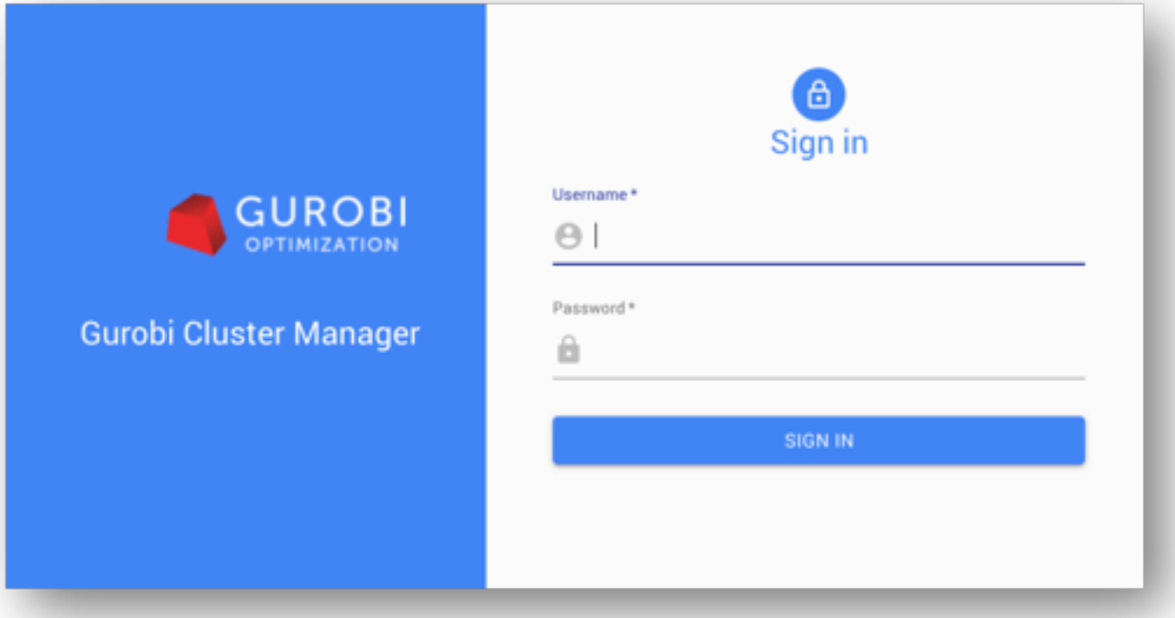
Runtime
9.5.0
Requested runtime to execute the job

Priority
0
Priority of the job

Cluster Manager and Compute Server facilitate the deployment and use of optimization services on-premises or on private cloud.

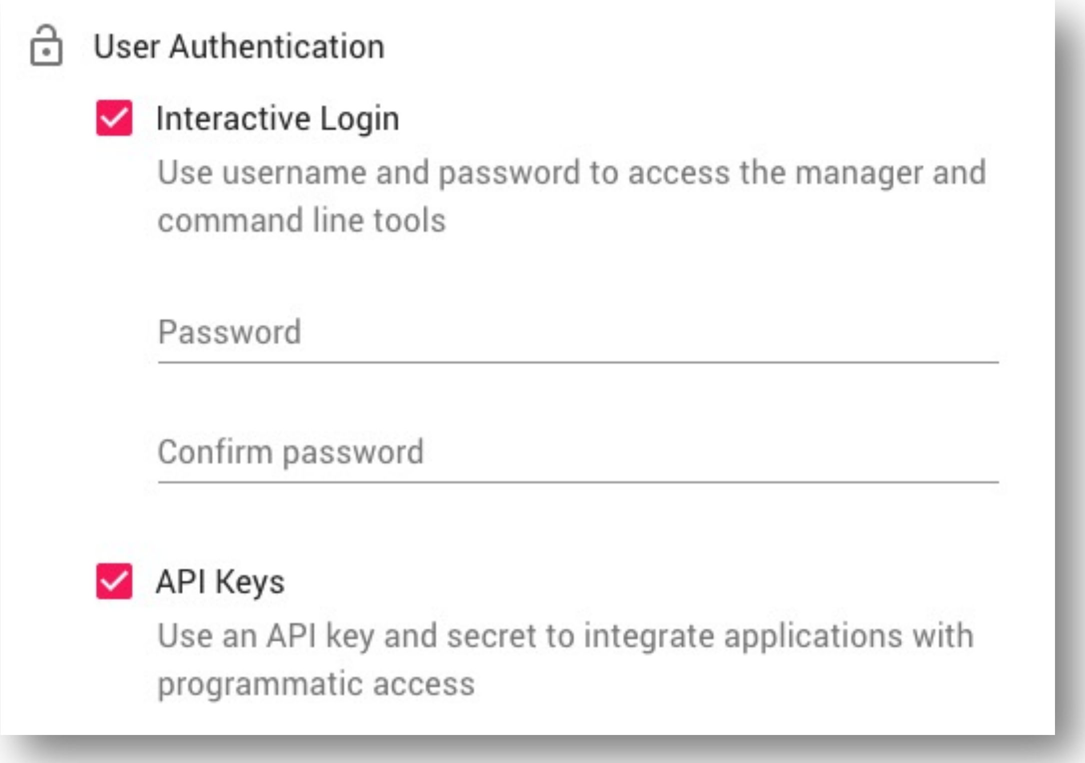
Improved Account Management

- System Administrator
  - Create/delete accounts
  - Assign role
  - Set/reset initial the password
  - Disable/enable account (New)
- User Roles
  - Sysadmin users
  - Standard users
  - Admin users (kill jobs, access licenses)
  - Read-only role (New)
- Default users (to delete before real deployment)
  - gurobi/pass
  - admin/admin
  - sysadmin/cluster



Improved Authentication

- Types of authentication
  - Interactive login for users (username/password)
  - API keys for applications (key/secret)
- Can now be configured separately (New)
  - *Monitoring*: use interactive login only
  - *System account*: use API keys only
  - *Developer*: use interactive login and API keys
- System administrator actions (New)
  - Set initial authentication type
  - Change authentication type



User Authentication

☑ Interactive Login

Use username and password to access the manager and command line tools

Password

Confirm password

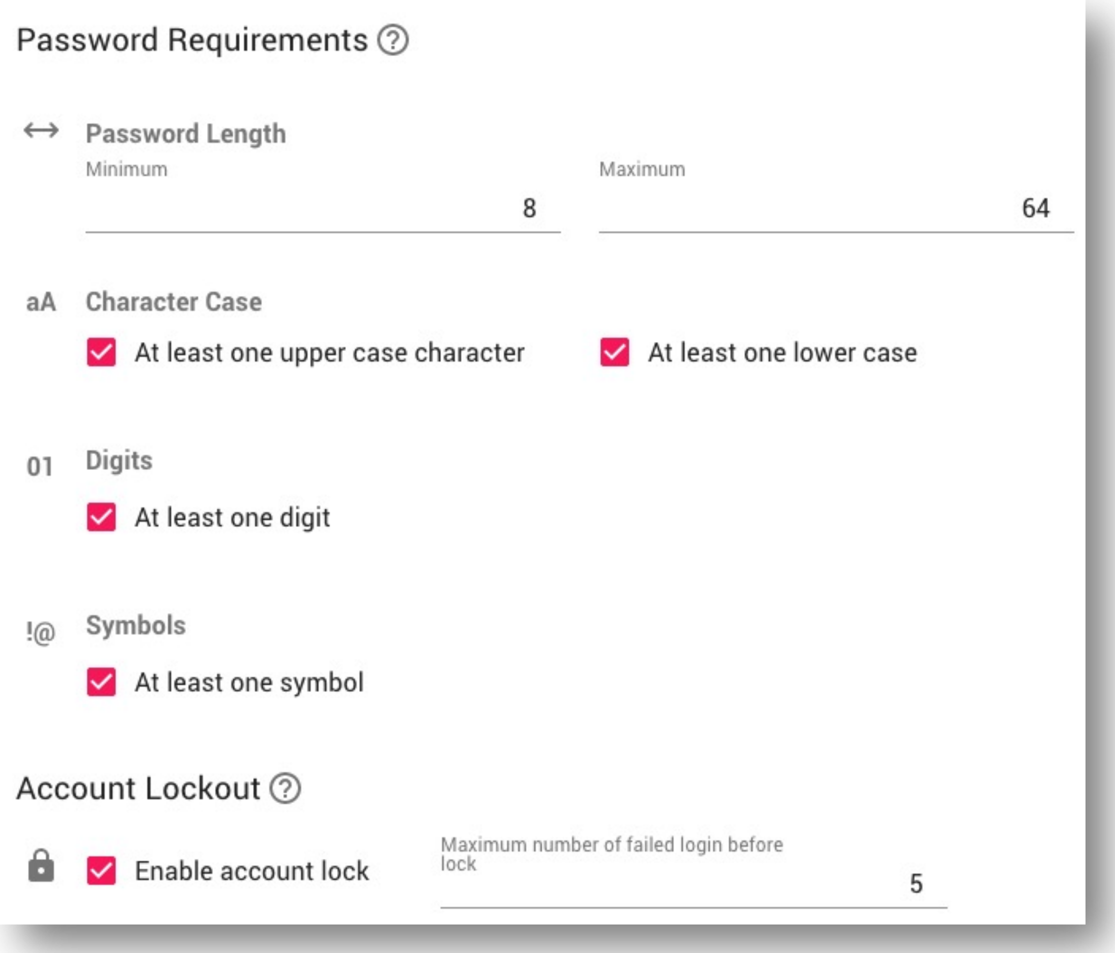
☑ API Keys

Use an API key and secret to integrate applications with programmatic access

Password Policy (New in v9.5)

- Password policy
  - Password requirements
  - Lockout behavior

- System administrator actions
  - Change the policy that will apply to new passwords
  - Unlock a user by resetting the password



Password Requirements

Password Length

Minimum: 8

Maximum: 64

Character Case

☑ At least one upper case character

☑ At least one lower case

Digits

☑ At least one digit

Symbols

☑ At least one symbol

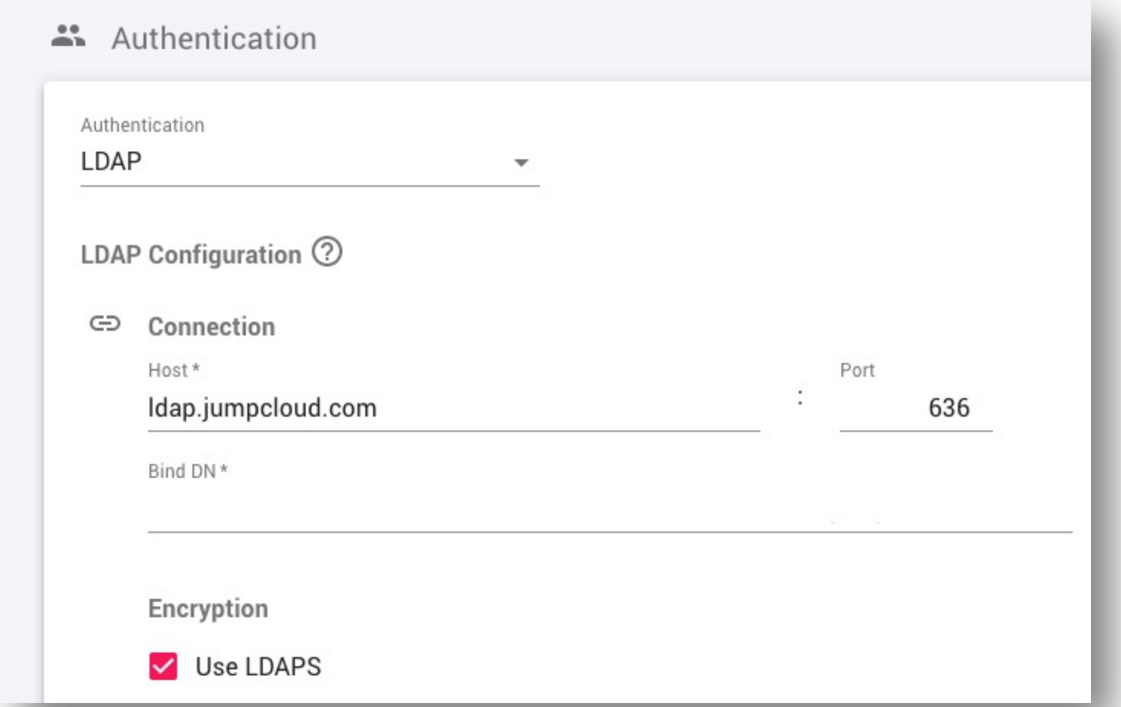
Account Lockout

☑ Enable account lock

Maximum number of failed login before lock: 5

LDAP Integration (New in v9.5)

- LDAP Server
  - Centralized user management
  - Users and Groups
  - Authentication
- LDAP Integration
  - Define who can access the Cluster Manager centrally in the LDAP server
  - Account will be created automatically during user login
  - Accounts are synchronized, users will be disabled if not a member any longer.
- Excluded accounts
  - *System administrators*: can administrate even if there is an issue with the LDAP
  - *System users* (API keys only): applications can continue to run even if there is an issue with the LDAP



Authentication

Authentication: LDAP

LDAP Configuration

Connection

Host *: ldap.jumpcloud.com

Port: 636

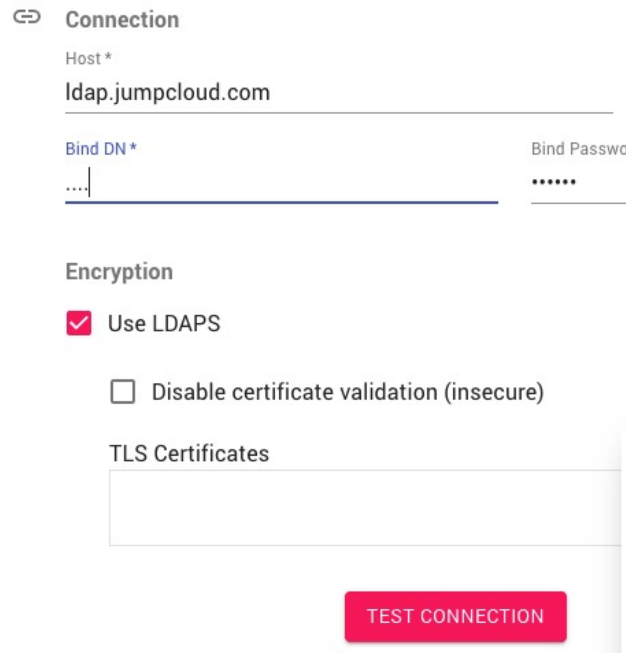
Bind DN *

Encryption

☑ Use LDAPS

LDAP Integration - Settings

- Connection to server
  - Host, port,
  - Bind DN, password
  - LDAPS, certificates
  - Test
- Filter and Mapping
  - Group of users
  - Mapping of attributes
  - Test
- Synchronization



Connection

Host *: ldap.jumpcloud.com

Port: 636

Bind DN *: ....

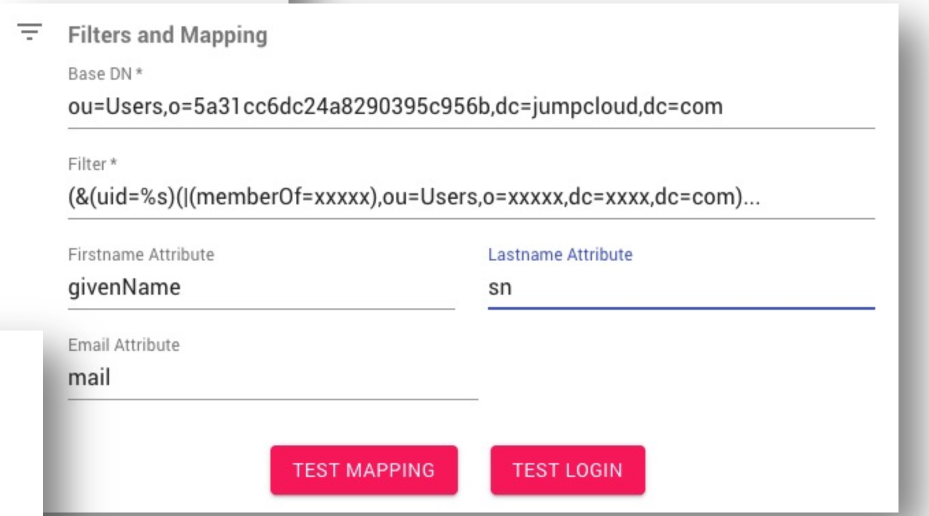
Bind Password: ••••••

Encryption

☑ Use LDAPS

□ Disable certificate validation (insecure)

TLS Certificates



Filters and Mapping

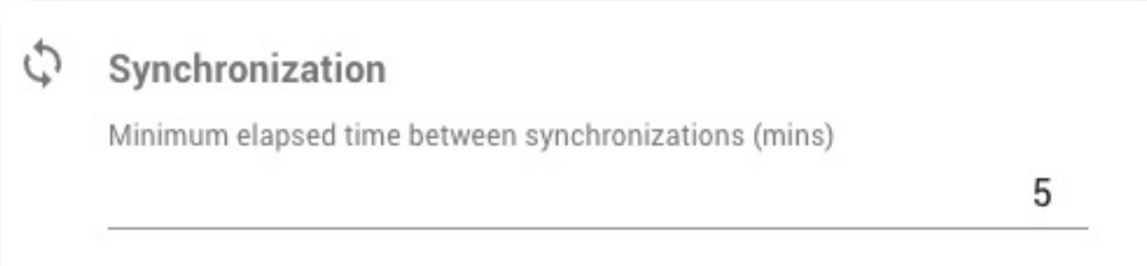
Base DN *: ou=Users,o=5a31cc6dc24a8290395c956b,dc=jumpcloud,dc=com

Filter *: (&(uid=%s)(|(memberOf=xxxxx),ou=Users,o=xxxxx,dc=xxxx,dc=com)...

Firstname Attribute: givenName

Lastname Attribute: sn

Email Attribute: mail

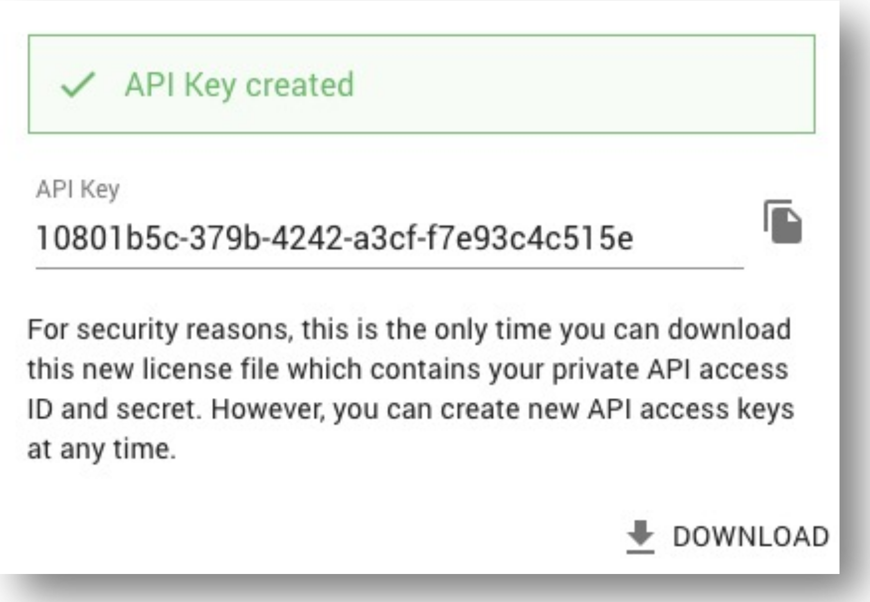


Synchronization

Minimum elapsed time between synchronizations (mins): 5

Improved API Key Management

- API keys for applications
  - Composed of an authentication token and secret key
  - Each user can manage their own API keys
  - Can be embedded into applications
- New Features in v9.5
  - Easier connection: client license file can now be downloaded when creating an API key
  - Better tracking of API keys: Application name, Description, Last timestamp, Last IP
  - API keys can be disabled/enabled

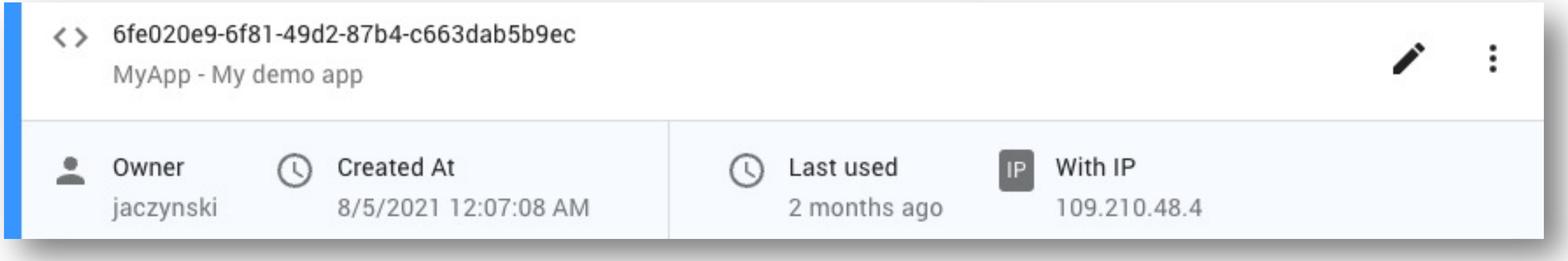


✓ API Key created

API Key

10801b5c-379b-4242-a3cf-f7e93c4c515e

For security reasons, this is the only time you can download this new license file which contains your private API access ID and secret. However, you can create new API access keys at any time.



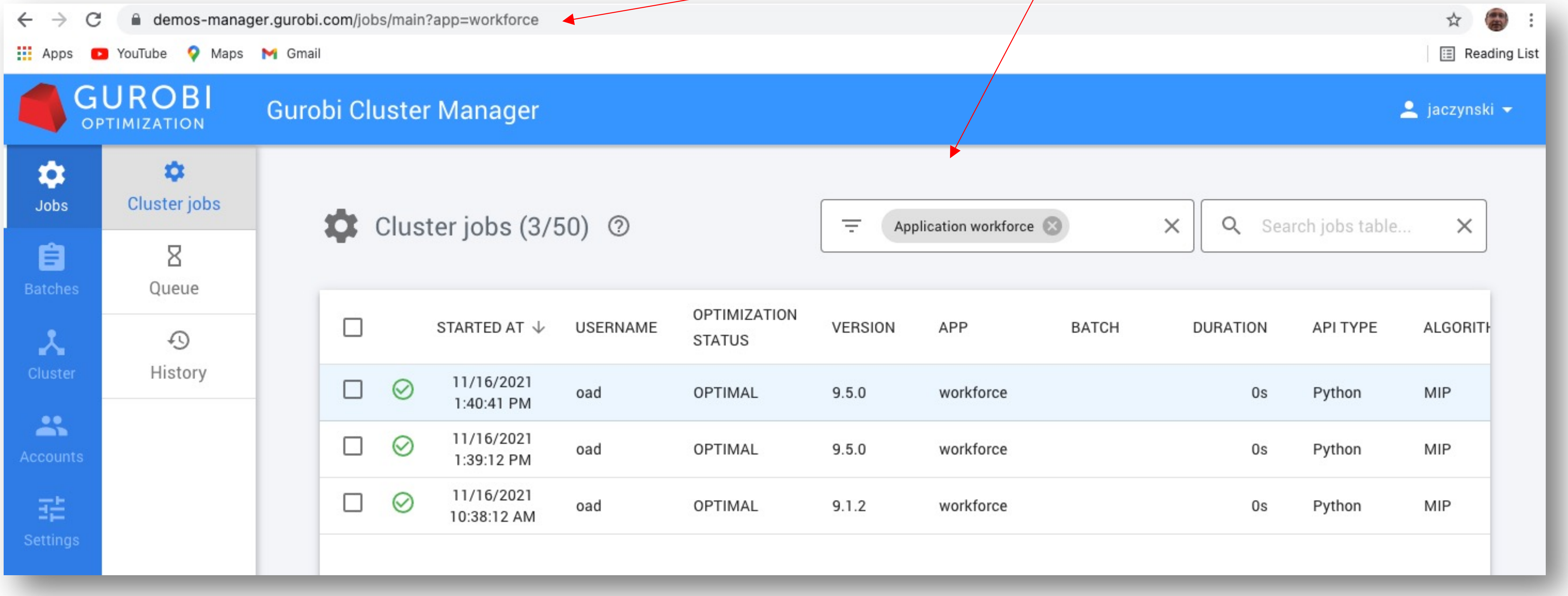
6fe020e9-6f81-49d2-87b4-c663dab5b9ec

MyApp - My demo app

Owner	Created At	Last used	With IP
jaczynski	8/5/2021 12:07:08 AM	2 months ago	109.210.48.4

Improved Filtering

Permalink (New)



demos-manager.gurobi.com/jobs/main?app=workforce

Cluster jobs (3/50)

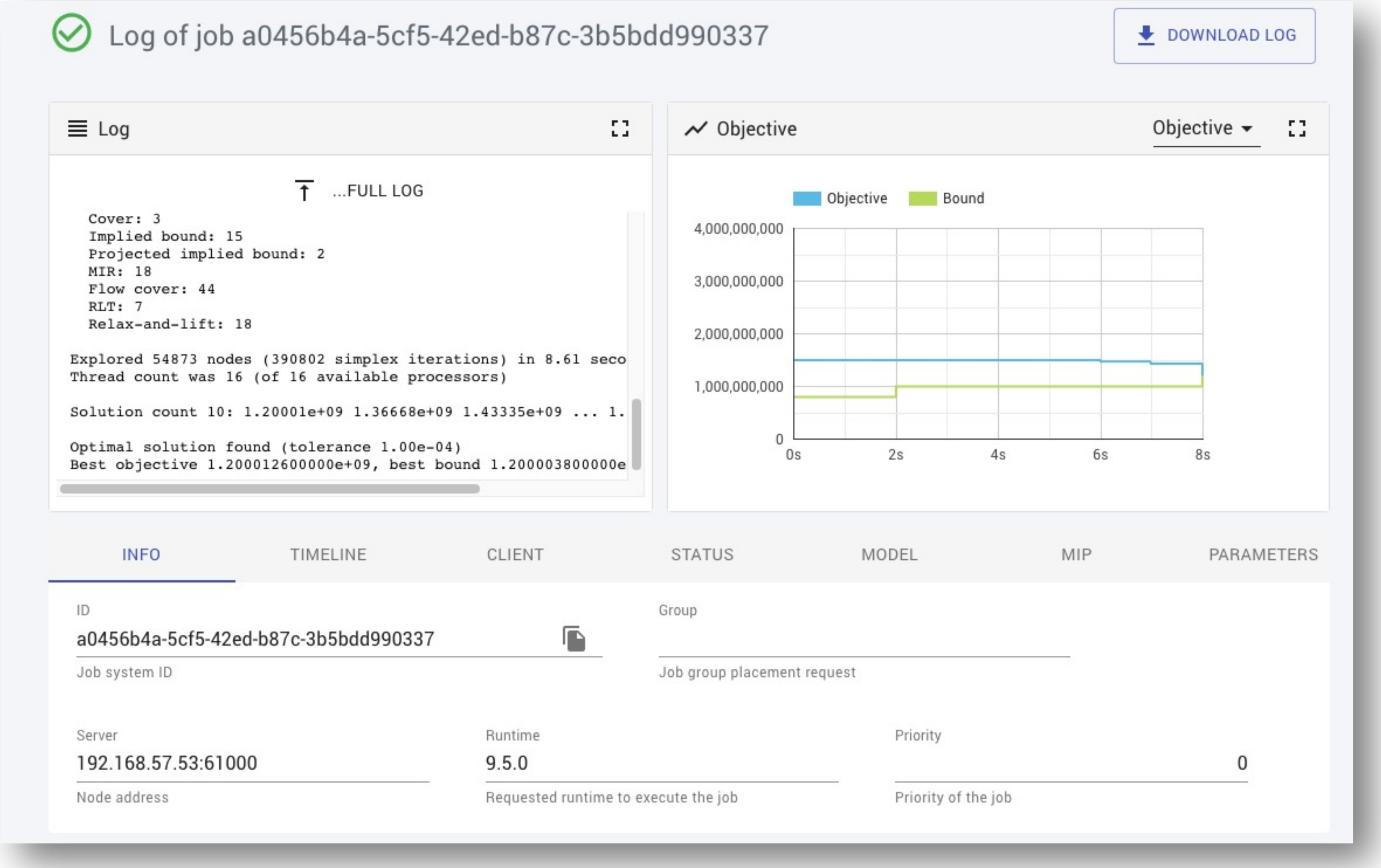
Application workforce

Search jobs table...

□		STARTED AT ↓	USERNAME	OPTIMIZATION STATUS	VERSION	APP	BATCH	DURATION	API TYPE	ALGORITH
□	✓	11/16/2021 1:40:41 PM	oad	OPTIMAL	9.5.0	workforce		0s	Python	MIP
□	✓	11/16/2021 1:39:12 PM	oad	OPTIMAL	9.5.0	workforce		0s	Python	MIP
□	✓	11/16/2021 10:38:12 AM	oad	OPTIMAL	9.1.2	workforce		0s	Python	MIP

Improved Job Dashboard

- Logs
  - Logs can be accessed live when the job is running
  - Logs are also archived in the history
- Job details
  - Info, timeline, status
  - Client
  - Model info
  - Parameters
- Metrics (New)
  - Objective, Gap



Log of job a0456b4a-5cf5-42ed-b87c-3b5bdd990337

...FULL LOG

```
   Cover: 3
   Implied bound: 15
   Projected implied bound: 2
   MIR: 18
   Flow cover: 44
   RLT: 7
   Relax-and-lift: 18

Explored 54873 nodes (390802 simplex iterations) in 8.61 seco
Thread count was 16 (of 16 available processors)

Solution count 10: 1.20001e+09 1.36668e+09 1.43335e+09 ... 1.

Optimal solution found (tolerance 1.00e-04)
Best objective 1.200012600000e+09, best bound 1.200003800000e
```

ID
a0456b4a-5cf5-42ed-b87c-3b5bdd990337
Job system ID

Group

Job group placement request

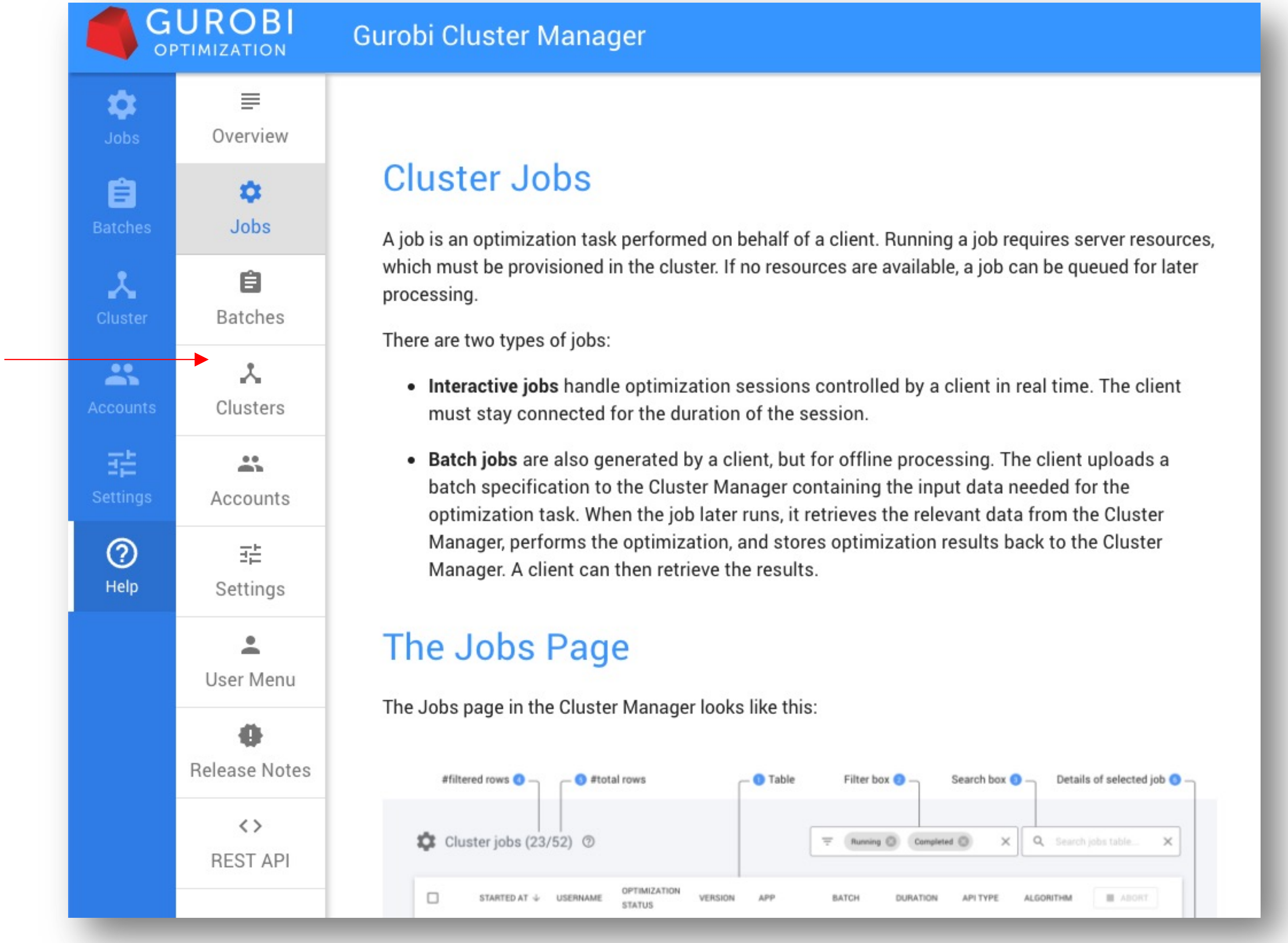
Server
192.168.57.53:61000
Node address

Runtime
9.5.0
Requested runtime to execute the job

Priority
0
Priority of the job

User Manual (New in v9.5)

User Manual Sections



Cluster Jobs

A job is an optimization task performed on behalf of a client. Running a job requires server resources, which must be provisioned in the cluster. If no resources are available, a job can be queued for later processing.

There are two types of jobs:

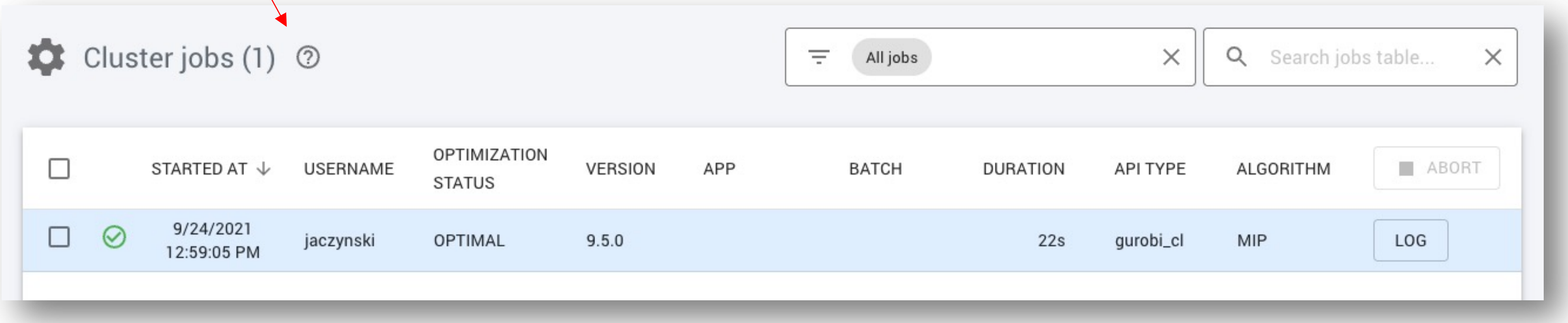
- **Interactive jobs** handle optimization sessions controlled by a client in real time. The client must stay connected for the duration of the session.
- **Batch jobs** are also generated by a client, but for offline processing. The client uploads a batch specification to the Cluster Manager containing the input data needed for the optimization task. When the job later runs, it retrieves the relevant data from the Cluster Manager, performs the optimization, and stores optimization results back to the Cluster Manager. A client can then retrieve the results.

The Jobs Page

The Jobs page in the Cluster Manager looks like this:

Contextual Help (New in v9.5)

Link to user manual



Cluster jobs (1) ⓘ

All jobs ✕

Search jobs table... ✕

□		STARTED AT ↓	USERNAME	OPTIMIZATION STATUS	VERSION	APP	BATCH	DURATION	API TYPE	ALGORITHM	ABORT
□	✓	9/24/2021 12:59:05 PM	jaczynski	OPTIMAL	9.5.0			22s	gurobi_cl	MIP	LOG

Improved Database Management

- Support of AWS DocumentDB 4.0 (New)
  - Easier deployment on AWS
  - https://aws.amazon.com/documentdb/

- Support for schema migration (New)
  - New features and schema changes can be applied automatically or manually when upgrading from 9.1.2

GUROBI OPTIMIZATION

Platform Enhancements in Gurobi 9.5

Compute Server/Cluster Manager

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Gurobi on Containerized Environments

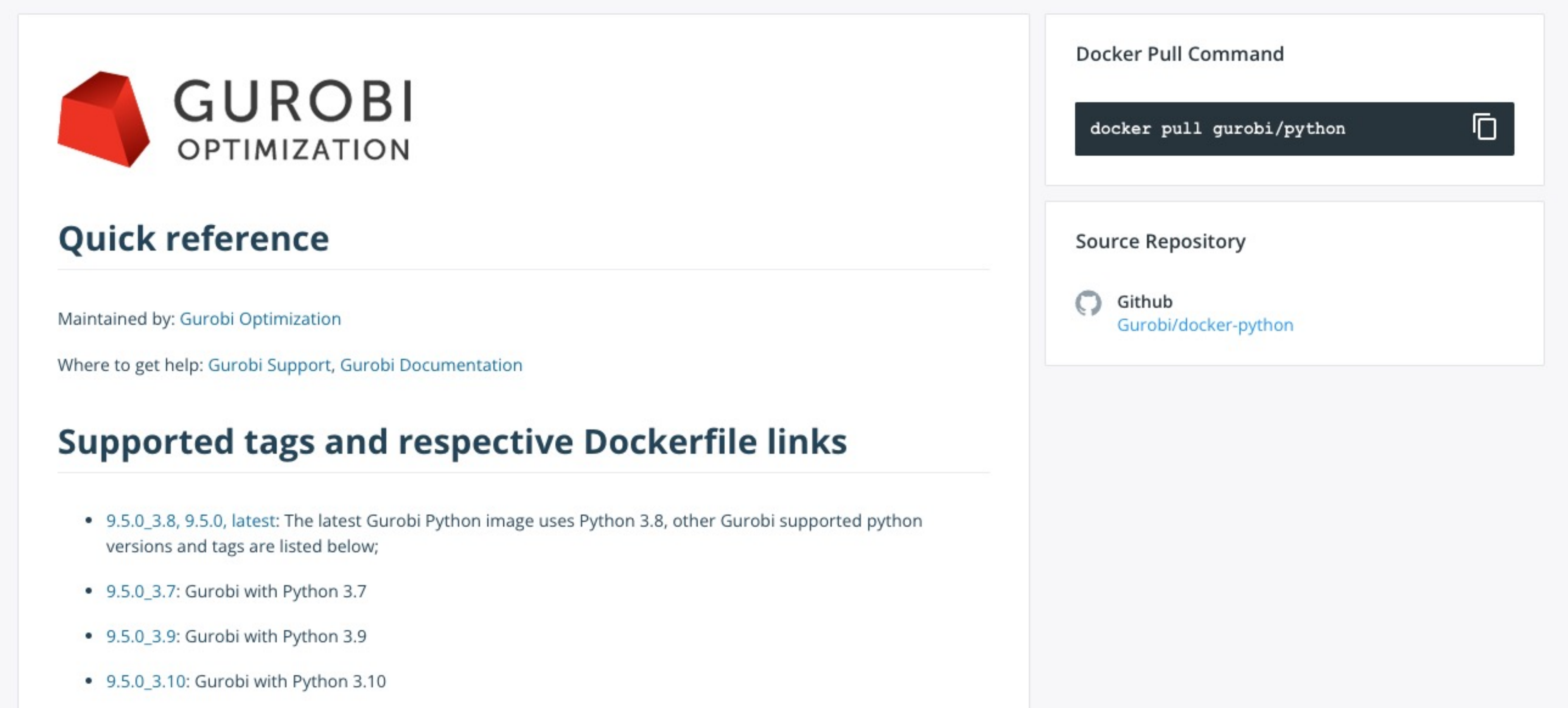
- Docker Hub Predefined Images
- Web License Service
- Best Practices for Kubernetes

Demo

Gurobi on Docker Hub

- Containerized environment usage is growing steadily
  - Build container images, and deploy in many environments (QA, Staging, Production)
  - Design, build and deploy micro-services architecture faster
  - Hosted solution on AWS, Microsoft Azure, Google, …
- Gurobi predefined images on Docker Hub
  - Gurobi Engine
    - gurobi/optimizer: Gurobi Optimizer (full distribution)
    - gurobi/python: Gurobi Optimizer (Python API gurobipy only)
      - Images available for Python 3.7, 3.8, 3.9 and 3.10 (New)
  - Compute server
    - gurobi/compute: Gurobi Compute Server
    - gurobi/manager: Gurobi Cluster Manager
  - Examples:
    - gurobi/python-example: Gurobi Optimizer example in Python (getting started)
    - gurobi/modeling-examples: Optimization modeling examples (Jupyter Notebook)

Docker Image Documentation



GUROBI OPTIMIZATION

Quick reference

Maintained by: Gurobi Optimization

Where to get help: Gurobi Support, Gurobi Documentation

Supported tags and respective Dockerfile links

- 9.5.0_3.8, 9.5.0, latest: The latest Gurobi Python image uses Python 3.8, other Gurobi supported python versions and tags are listed below;
- 9.5.0_3.7: Gurobi with Python 3.7
- 9.5.0_3.9: Gurobi with Python 3.9
- 9.5.0_3.10: Gurobi with Python 3.10

Docker Pull Command

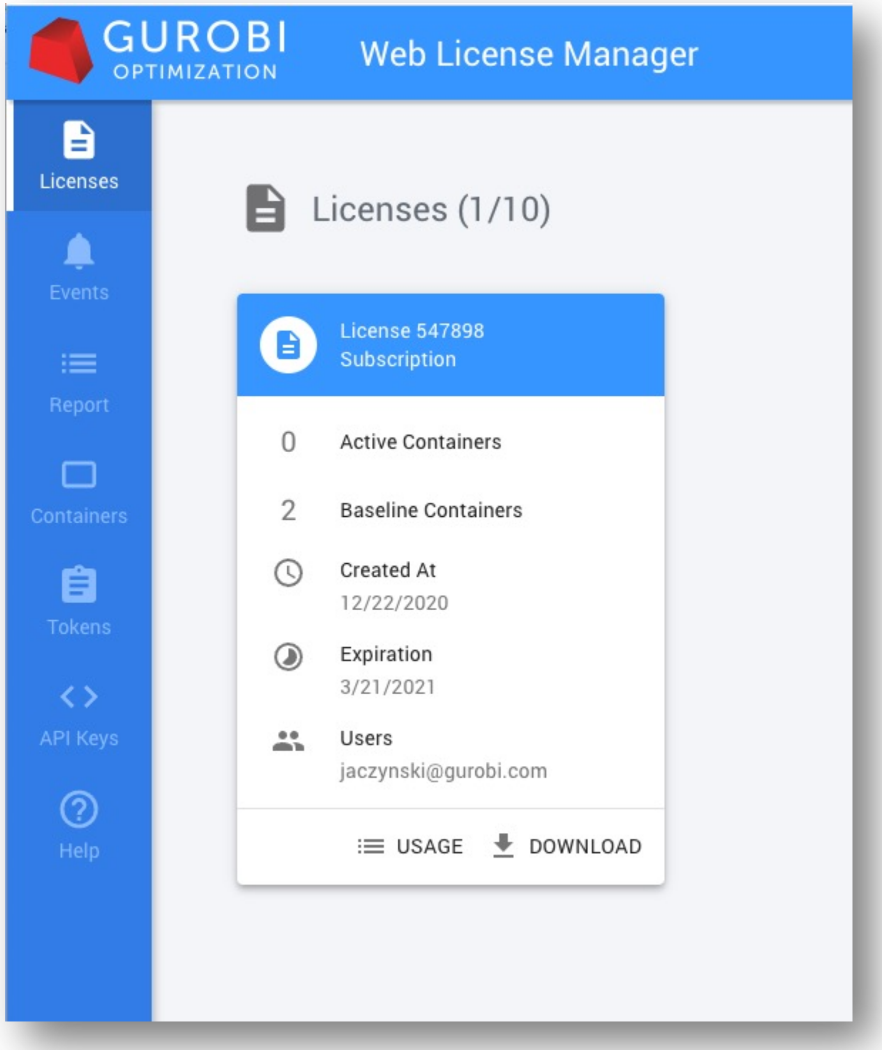
```
docker pull gurobi/python
```

Source Repository

Github
Gurobi/docker-python

Gurobi Web License Service (WLS)

- New licensing technology
  - Nothing to install
  - Servers are running in several regions worldwide
  - Dynamically activate use of Gurobi in containers
- License Manager
  - https://license.gurobi.com/
  - Academic can create licenses for free
  - Evaluation upon requests
  - Contact Sales for commercial subscription
- Supported platforms to deploy images
  - AWS: EKS (Kubernetes), ECS, Batch
  - Azure: AKS (Kubernetes), Container Instance, Batch
  - GCP: GKE (Kubernetes)
  - Red Hat OpenShift (Kubernetes)



Licenses (1/10)

License 547898
Subscription

0 Active Containers

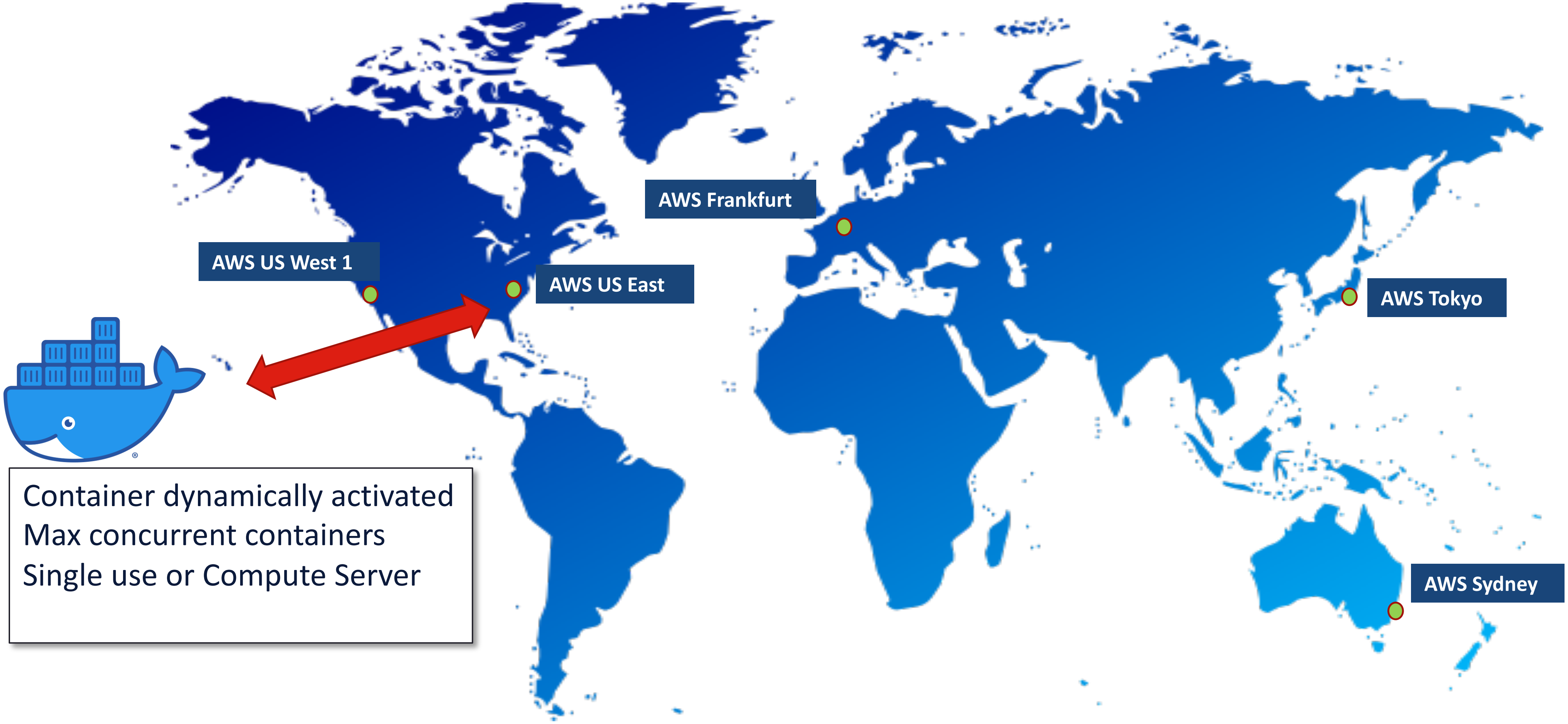
2 Baseline Containers

Created At
12/22/2020

Expiration
3/21/2021

Users
jaczynski@gurobi.com

Web License Service – Worldwide Availability

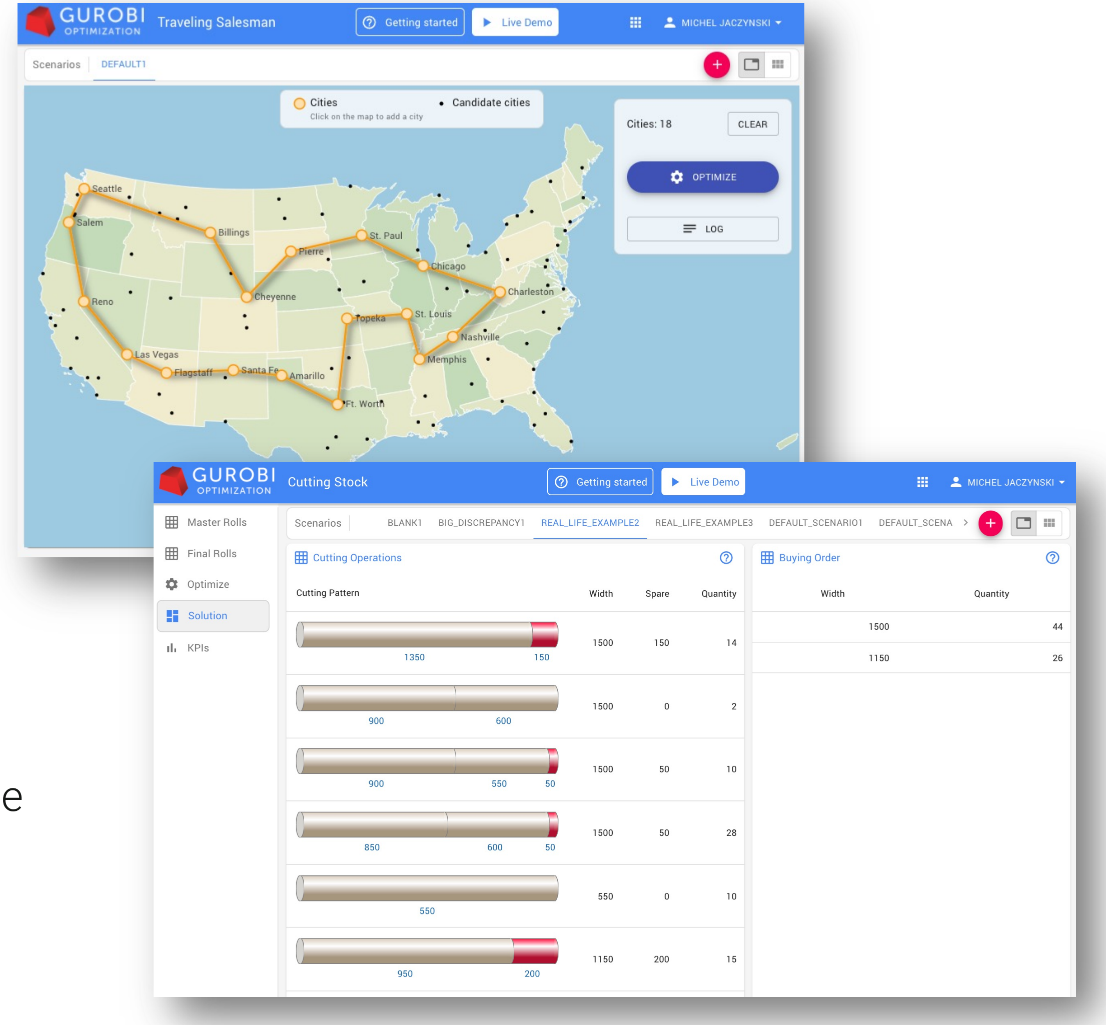


Container dynamically activated
Max concurrent containers
Single use or Compute Server

Gurobi Demos (http://demos.gurobi.com)

- Gurobi Optimization Application Demos (OAD)
  - Cutting Stock
  - Workforce assignment
  - Resource matching
  - Cell tower placement
  - Offshore wind farming
  - Facility location
  - TSP

- Deployment
  - Kubernetes cluster on AWS (EKS)
  - Using Cluster Manager/Compute server to execute optimization
  - Web License Service



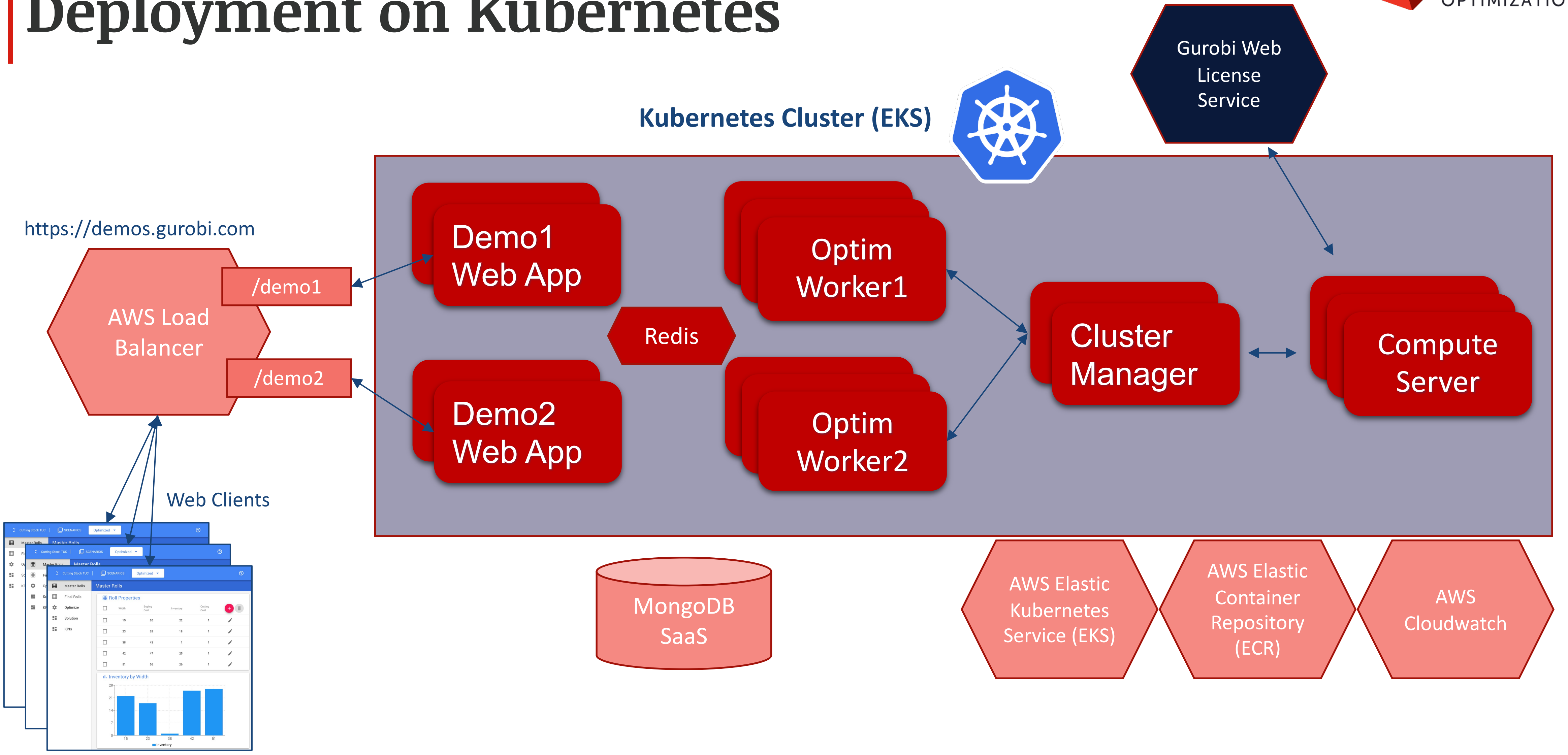
Cutting Operations

Cutting Pattern	Width	Spare	Quantity
1350 150	1500	150	14
900 600	1500	0	2
900 550 50	1500	50	10
850 600 50	1500	50	28
550	550	0	10
950 200	1150	200	15

Buying Order

Width	Quantity
1500	44
1150	26

Deployment on Kubernetes



Kubernetes Cluster (EKS)

Gurobi Web License Service

https://demos.gurobi.com

AWS Load Balancer

/demo1

/demo2

Demo1 Web App

Demo2 Web App

Redis

Optim Worker1

Optim Worker2

Cluster Manager

Compute Server

Web Clients

MongoDB SaaS

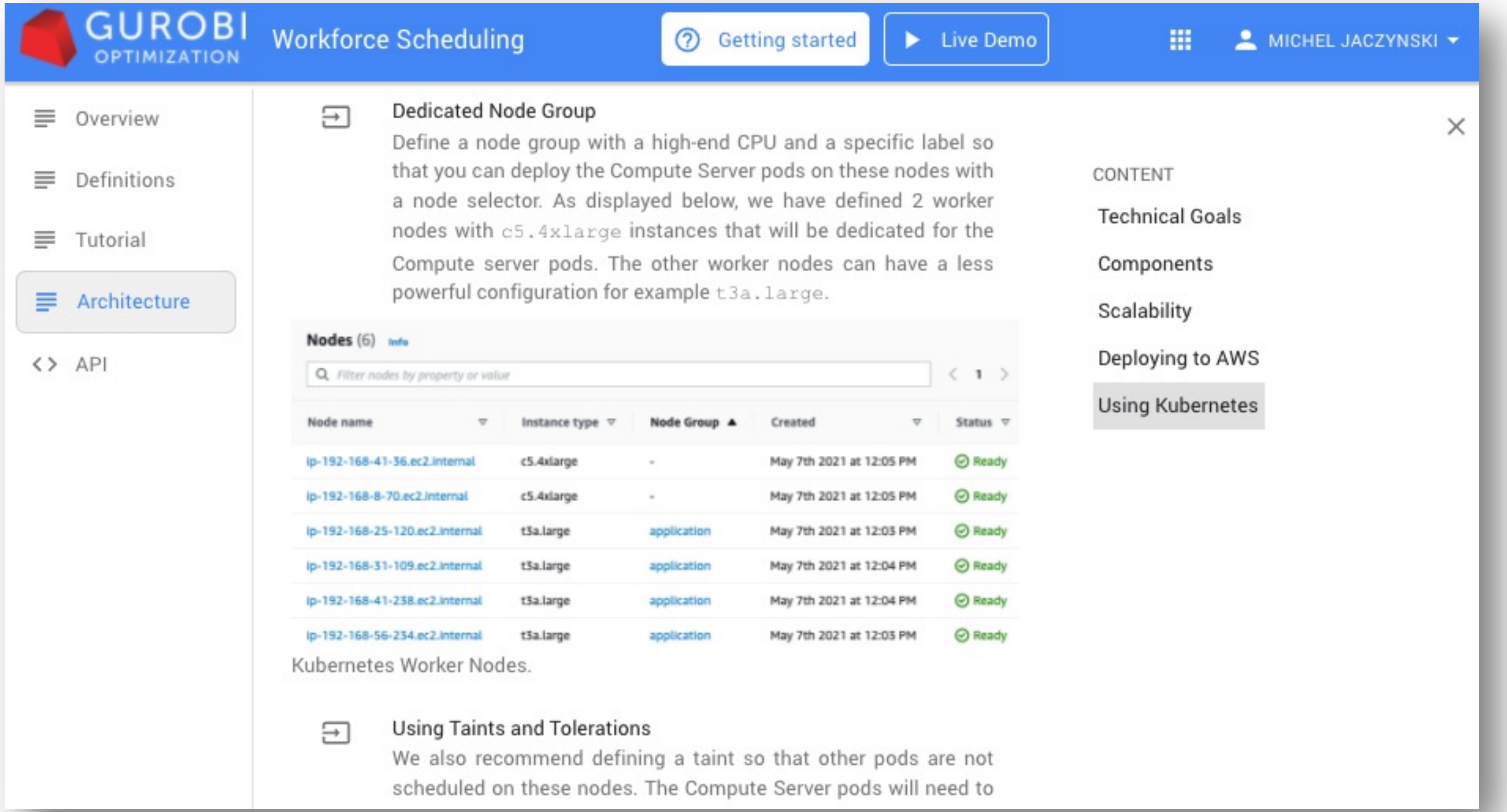
AWS Elastic Kubernetes Service (EKS)

AWS Elastic Container Repository (ECR)

AWS Cloudwatch

Documentation and Best Practices

https://demos.gurobi.com/workforce/doc/architecture



Dedicated Node Group

Define a node group with a high-end CPU and a specific label so that you can deploy the Compute Server pods on these nodes with a node selector. As displayed below, we have defined 2 worker nodes with `c5.4xlarge` instances that will be dedicated for the Compute server pods. The other worker nodes can have a less powerful configuration for example `t3a.large`.

Nodes (6) Info

Filter nodes by property or value

Node name	Instance type	Node Group ▲	Created	Status
ip-192-168-41-36.ec2.internal	c5.4xlarge	-	May 7th 2021 at 12:05 PM	✓ Ready
ip-192-168-8-70.ec2.internal	c5.4xlarge	-	May 7th 2021 at 12:05 PM	✓ Ready
ip-192-168-25-120.ec2.internal	t3a.large	application	May 7th 2021 at 12:03 PM	✓ Ready
ip-192-168-31-109.ec2.internal	t3a.large	application	May 7th 2021 at 12:04 PM	✓ Ready
ip-192-168-41-238.ec2.internal	t3a.large	application	May 7th 2021 at 12:04 PM	✓ Ready
ip-192-168-56-234.ec2.internal	t3a.large	application	May 7th 2021 at 12:03 PM	✓ Ready

Kubernetes Worker Nodes.

Using Taints and Tolerations

We also recommend defining a taint so that other pods are not scheduled on these nodes. The Compute Server pods will need to

GUROBI
OPTIMIZATION

Platform Enhancements in Gurobi 9.5

Compute Server/Cluster Manager

- Improved account management
- Integration with LDAP repositories for centralized user management
- Improved API key management
- Improved usability
- Support of Amazon Web Services DocumentDB 4.0 database

Gurobi on Containerized Environments

- Docker Hub Predefined Images
- Web License Service
- Best Practices for Kubernetes

Demo

Demo Script

- Overview of demos:
  - TSP demo
  - Click optimize
- Overview of the manager
  - cluster
  - User management
  - Show jobs, logs, dashboard
  - User manual, contextual help
- Run a Job
  - Create API key, download
  - Run glass4
  - Show job, log and dashboard live

GUROBI
OPTIMIZATION

Thank You

For more information: gurobi.com